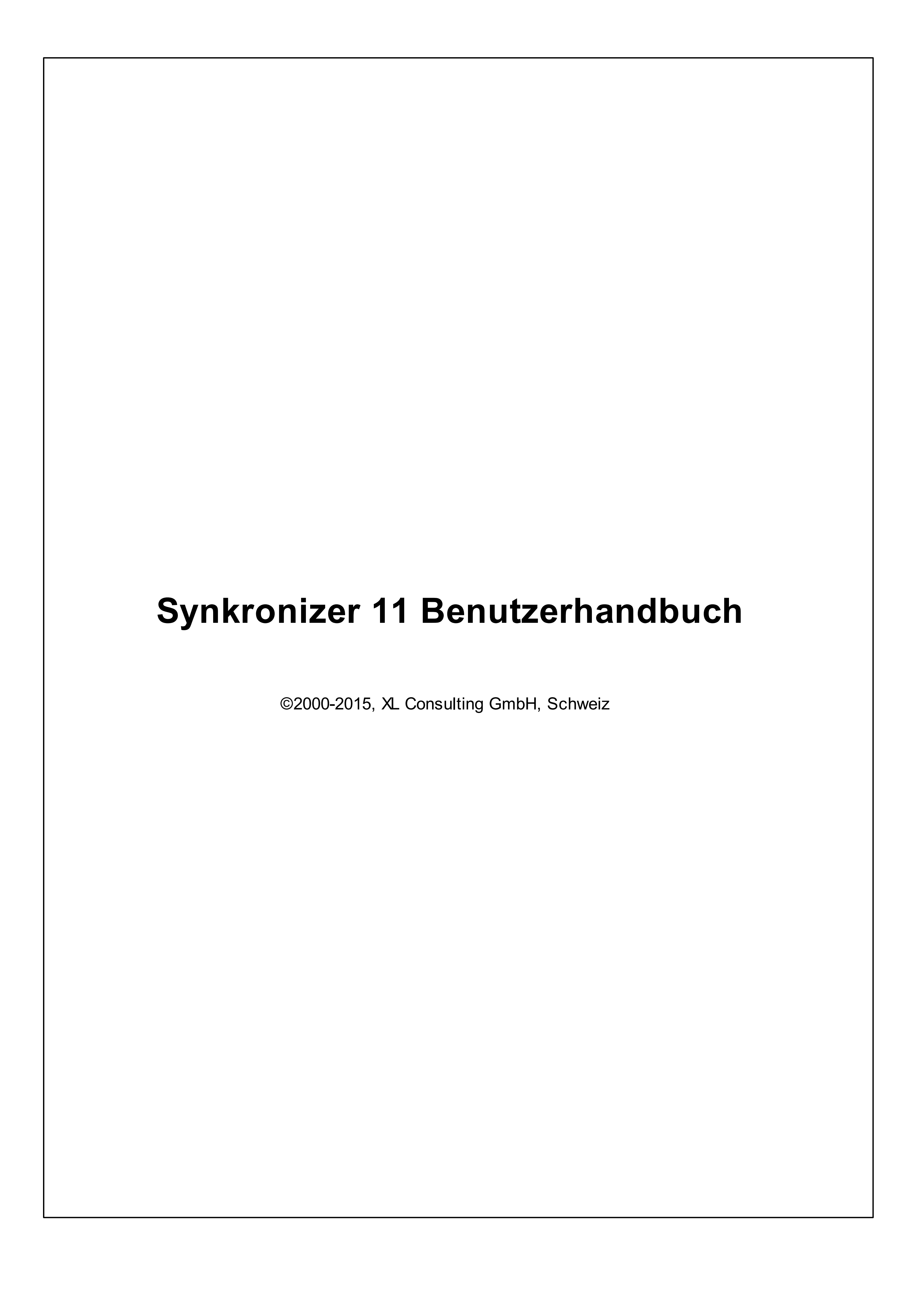


Synkronizer 11 Benutzerhandbuch

Table of Contents

Part I

1 Willkommen

Sehr geehrter Synkronizer-Benutzer,

Schön, dass Sie sich entschieden haben, mit Synkronizer 11 zu arbeiten. Synkronizer 11 ist das leistungsstärkste Excel-Vergleichsprogramm auf dem Markt. Synkronizer 11 ist das einzige Programm weltweit, das normale Excel-Dateien und Excel-Datenbanken vergleichen und aktualisieren kann.

Mit Synkronizer 11 können Sie folgende Arbeiten erledigen:

- Vergleichen von Excel-Dateien
- Aktualisieren/Transferieren von Differenzen
- Markieren der Differenzen in den zugrundeliegenden Tabellen
- Erstellen eines Abweichungsprotokolls
- Unterschiedliche Excel-Tabellen bzw. -Datenbanken fusionieren
- Auffinden von Diskrepanzen/Duplikaten
- Vergleichsprozesse mittels Projekten automatisieren
- Syntax für Kommandozeilen und Visual Basic (VBA)-Anwendungen

Dieses Manual führt Sie in die Synkronizer 11 Umgebung ein und beschreibt und erklärt die wesentlichen Funktionen.

Natürlich ist Synkronizer 11 nicht perfekt. Wir sind immer daran interessiert von unseren Anwendern zu lernen - also von Ihnen. Kontaktieren Sie uns direkt, falls Sie Anregungen haben, zum Beispiel zu Menüpunkten, Einstellungen, Formularen oder zu diesem Manual.

Mit freundlichen Grüssen

Thomas Strübi & Jurgen Volkerink
XL Consulting GmbH

1.1 Was ist neu?_2

Wenn Sie bereits ein Anwender von Synkronizer sind, dann kennen Sie das Programm bestens, dennoch beinhaltet Synkronizer 11 ein paar Neuerungen, die wir Ihnen gerne vorstellen möchten:

- Synkronizer 11 ist ein COM Add-in
- Das Programm funktioniert mit folgenden Excel-Versionen:

Excel 2013 / 32-Bit und 64-Bit
Excel 2010 / 32-Bit und 64-Bit
Excel 2007 / 32-Bit

- folgende Betriebssysteme werden unterstützt:
  Windows 8 / 32-Bit und 64-Bit
  Windows 7 / 32-Bit und 64-Bit
  Windows Vista / 32-Bit und 64-Bit
  Windows XP / 32-Bit und 64-Bit
  Windows Server 2012
  Windows Server 2008

- Das Programm hat eine neue Benutzeroberfläche.

- Neu können Kommentare verglichen werden.

- Excel-Namen können verglichen werden.

- Es gibt neu eine Differenzkategorie "Eingegebene Werte & Formeln" und "Kalkulierte Werte". Auf diese Weise werden Zelldifferenzen genauer identifiziert. Bei der Kategorie "Eingegebene Werte & Formeln" werden unterschiedliche Zellwerte oder Formeln ausgewiesen. Beim Differenztyp "Kalkulierte Werte" ist die Formel identisch, aber das Ergebnis der Kalkulation ist unterschiedlich.

- Neue Filterfunktionen.

- Aufruf per Kommandozeile. Mit der Developer-Edition kann man den Synkronizer per Kommandozeile steuern. Die Developer-Edition wird nicht vor **Sommer 2014** lieferbereit sein.

1.2 Editionen_2

Es gibt insgesamt drei verschiedene Synkronizer Editionen. Der Funktionsumfang dieser Editionen ist in der nachfolgenden Tabelle beschrieben:

	Professional	**Developer**	**Trial** *
Gesamter Zellbereich *	•	•	
Differenzen markieren	•	•	•
Differenzprotokoll erstellen	•	•	•
Differenzen transferieren	•	•	•
Resultatebildschirm	•	•	•
Filter verwenden	•	•	•
Zellwerte vergleichen	•	•	•
Zellformeln vergleichen	•	•	•
Zellkommentare vergleichen	•	•	•
Textformate vergleichen	•	•	•
Zellformate vergleichen	•	•	•
Excel-Namen vergleichen	•	•	•
Differenzen gliedern	•	•	•
Mehrere Tabellen vergleichen	•	•	•

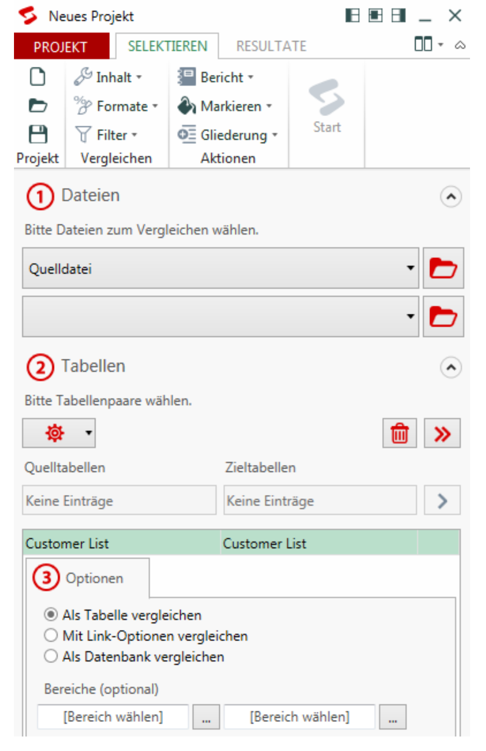
Datenbankvergleich	•	•	•
Projekte verwenden	•	•	•
Aufruf per Kommandozeile		•	
VBA Zugriff		•	
Preis	EUR 79.--	EUR 149.--	Gratis

* Mit der Trial-Edition kann nur der Zellbereich A1:Z100 verglichen werden.

Part II

2 Selektieren Register

Beim Aufruf des Synkronizers wird das Hauptformular oder Selektieren-Register angezeigt. Das Register dient zum Selektieren von ganzen Projekten oder der Auswahl zweier Excel-Dateien, die verglichen werden sollen, zudem können praktisch alle Vergleichsoptionen, Filter und Aktionen in diesem Register definiert werden.



Symbolleiste

In diesem Bereich sind verschiedene Befehle verfügbar. Sie können Projekte verwalten, Vergleichseinstellungen vornehmen und die Aktionen definieren.

Dateien wählen

In diesem Bereich werden die Dateien gewählt.

Tabellen wählen

Hier können Sie die folgenden Tabellenoptionen festelegen:
- Tabellentypen (geschützt, verborgen, usw.)
- Tabellen wählen und löschen
- Tabellen manuell wählen

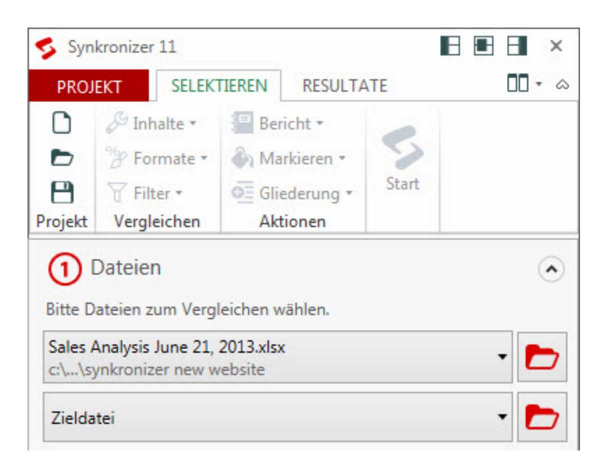
Tabellen Optionen

Folgende Optionen können festgelegt werden:
- Bereiche
- Datenbankoptionen
- Tabellen 1:1 verbinden

Die einzelnen Einstellungen werden in den weiteren Kapiteln beschrieben.

Anmerkung
Alle leeren Arbeitsmappen und alle Arbeitsmappen mit geschützten Fenstern werden geschlossen, wenn das Formular startet wird.

2.1 Dateien



In diesem Bereich können Sie bestimmen, welche Dateien miteinander verglichen werden sollen, dazu stehen Ihnen zwei Eingabefelder zur Verfügung. Im oberen Feld wird die erste Datei und im unteren Feld die zweite Datei eingetragen.

Mit den Symbolen "Datei öffnen" werden die Excel-Dateien gewählt. Alternativ können die Dateien auch mit den DropDown-Schaltflächen gewählt werden. Wenn Sie diese Schaltfläche klicken, erscheint eine Liste mit den kürzlich geöffneten Excel-Dateien.

Projekte

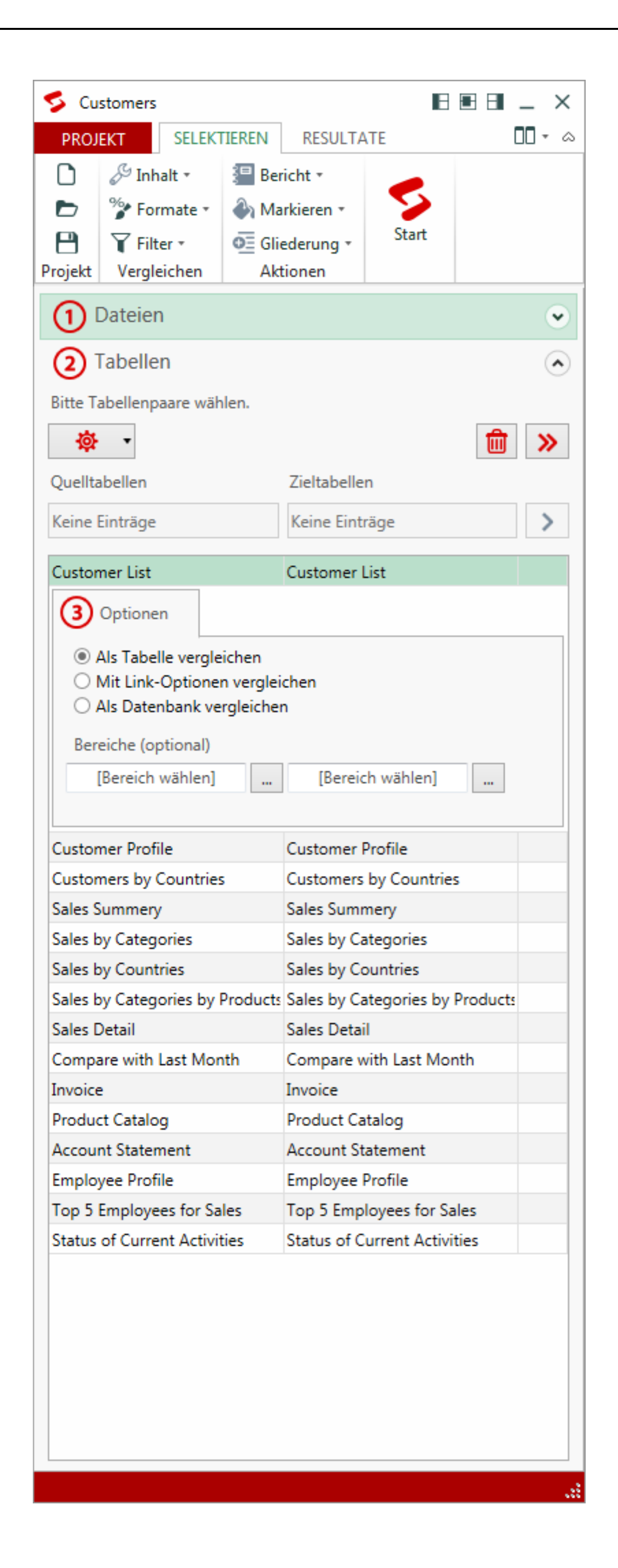
Falls Sie die Dateien in einem Projekt gespeichert haben, können die Dateien auch mittels der Projektbefehle geöffnet werden. Genauere Informationen dazu entnehmen Sie dem Kapitel Projekte.

Unterstützte Dateitypen

Sämtliche Dateien, die Microsoft Excel unterstützt, können mit dem Synkronizer geöffnet werden.

2.2 Tabellen

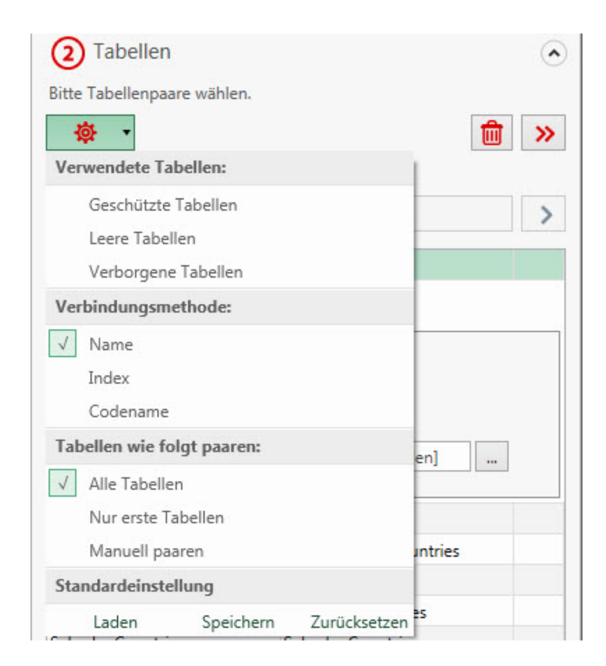
Im Bereich Tabellen werden die Tabellen geladen, die verglichen werden sollen. Zudem werden in diesem Bereich tabellenspezifische Einstellungen wie Bereiche, Datenbank- oder Vergleichsoptionen definiert.


Customers
PROJEKT
SELEKTIEREN
RESULTATE
Inhalt
Formate
Filter
Bericht
Markieren
Gliederung
Start
Projekt
Vergleichen
Aktionen
1 Dateien
2 Tabellen
Bitte Tabellenpaare wählen.
Quelltabellen
Zieltabellen
Keine Einträge
Keine Einträge
Customer List
Customer List
3 Optionen
Als Tabelle vergleichen
Mit Link-Optionen vergleichen
Als Datenbank vergleichen
Bereiche (optional)
[Bereich wählen]
[Bereich wählen]
Customer Profile
Customers by Countries
Sales Summery
Sales by Categories
Sales by Countries
Sales by Categories by Products
Sales Detail
Compare with Last Month
Invoice
Product Catalog
Account Statement
Employee Profile
Top 5 Employees for Sales
Status of Current Activities

2.2.1 Einstellungen

Tabellen automatisch paaren

Sobald Sie zwei Dateien gewählt haben, versucht der Synkronizer automatisch die Tabellen zu paaren. Die Paarungs-Einstellungen werden mit dem "Tabellenpaarungs"-Element festgelegt:



Folgende Optionen können in diesem Element festgelegt werden:

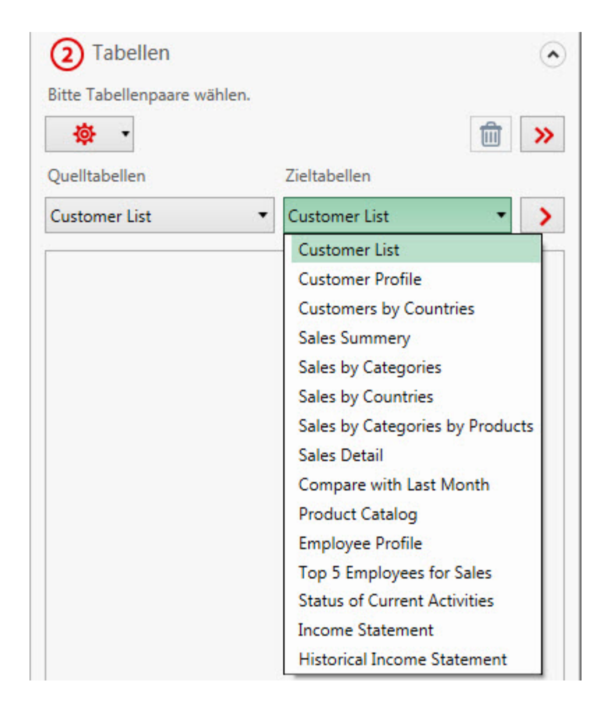
- **Verwendete Tabellen**
  Diese Tabellentypen werden zusätzlich gelistet.

- **Verbindungsmethode**
  Je nach Einstellung werden die Tabellen nach deren Namen, Index (Position innerhalb der Datei) oder Codename (interner Tabellenname) verbunden.

- **Tabellen wie folgt paaren**
  Hier können Sie bestimmen, wie die Tabellen gepaart werden. Bei "alle Tabellen" werden sämtliche gleichnamigen Tabellenpaare in die Tabellenliste übertragen, bei "nur erste Tabellen" hingegen nur die ersten Tabellen. Wenn die Tabellen nicht automatisch gepaart werden sollen, kann man "Manuell paaren" einstellen.

Tabellen manuell paaren

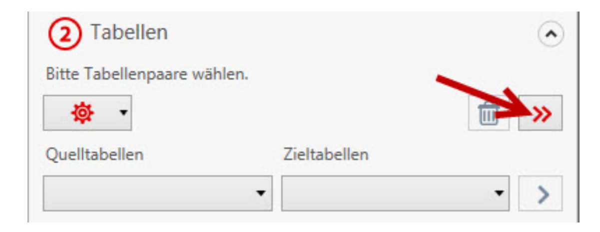
Die Tabellen können auch manuell gepaart werden. Dies geschieht in der Kopfzeile des Tabellen-Listenfeldes:



Oberhalb der Tabellenliste befinden sich zwei DropDown-Felder (siehe Grafik). Mittels dieser Felder können die gewünschten Tabellen gewählt werden. Nun muss nur noch die Selektion mit der Pfeiltaste rechts bestätigt werden und das Tabellenpaar wird in die Liste aufgenommen.

Alle Tabellen paaren

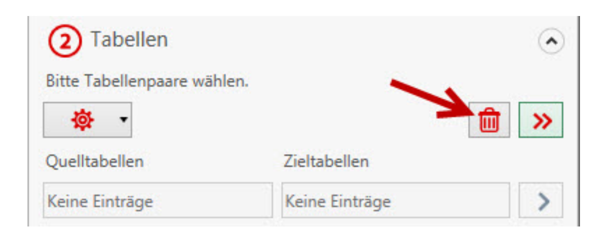
Sie können auch alle Tabellen paaren. Dies geschieht mit der folgenden Schaltfläche:



Es werden dann alle Tabellen mit der selben Bezeichnung gepaart.

Tabellenpaare entfernen

Wenn Sie nicht alle Tabellenpaare vergleichen wollen, können Sie die Tabellenliste mit der **"Löschen"-Taste** leeren.

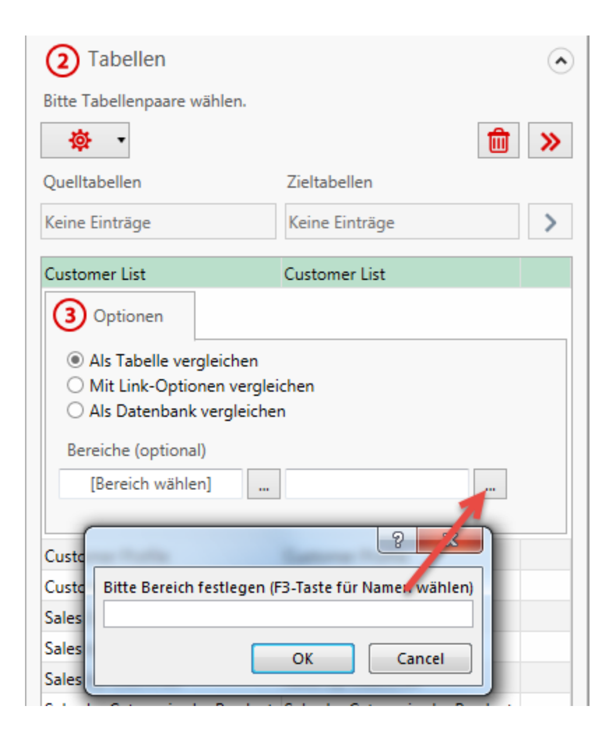


Alternativ können die Tabellenpaare entfernt werden, indem Sie das unerwünschte Tabellenpaar wählen und die "Löschen"-Taste klicken.

Keine Tabellen vorhanden/gefunden!

Wenn keine Tabellen in den DropDown-Listen vorhanden sind, dann sind Ihre Tabellen entweder geschützt, leer und/oder verborgen. In diesem Fall müssen Sie den Anzeigemodus für Tabellen ändern. Genauere Informationen finden Sie oben im Kapitel "Tabellen automatisch paaren":

2.2.2 Bereiche



Falls Sie nicht die ganze Tabelle vergleichen möchten, können Sie den gewünschten Bereich speziell definieren. Gehen Sie wie folgt vor:

1. Klicken Sie mit der Maus auf das Tabellenpaar.
2. Unter dem Tabellenpaar erscheint der Optionsbereich "3 Optionen". Nun können Sie mittels der Bereichs-Schaltflächen den gewünschten Bereich festlegen.
3. In der Tabellenliste wird zusätzlich der Bereich angezeigt.

Anmerkung:
Benannte Bereiche können ebenfalls eingegeben werden. Multiarea Bereiche dürfen hingegen nicht verwendet werden.

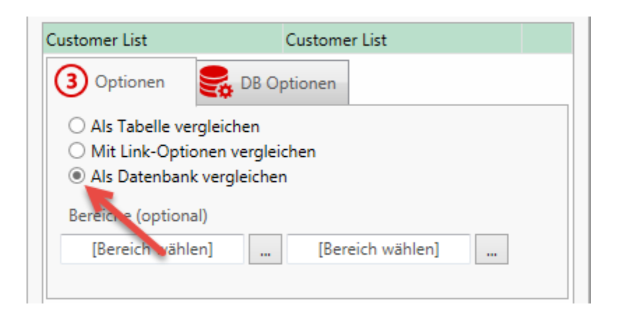
2.2.3 Datenbanken

Wenn Ihre Tabellen eine Datenbankstruktur enthalten, dann empfiehlt sich ein Datenbankvergleich. Sollten Sie nicht genau wissen, was eine Datenbank ist und wie sie funktioniert, dann lesen Sie zuerst das Kapitel "Allgemeine Informationen über Datenbanken" durch.

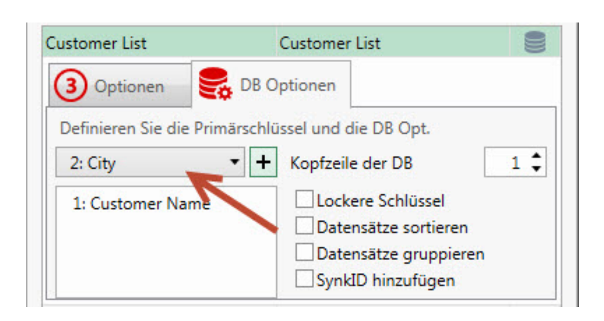
Gehen Sie wie folgt vor, um die Datenbankoptionen einzugeben:

1. Wählen Sie das Tabellenpaar, das eine Datenbankstruktur enthält.

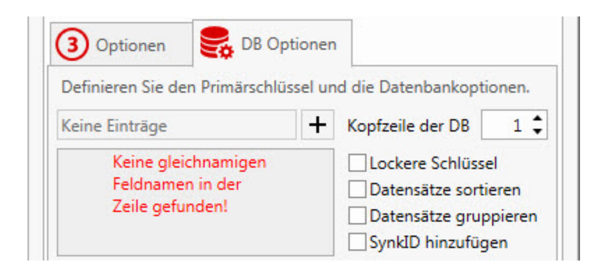
2. Klicken Sie mit der Maus auf das Kontrollkästchen "Als Datenbank vergleichen".



4. Das Register "DB Optionen" erscheint. Nun müssen Sie den Primärschlüssel bilden, anhand dessen die Datenbanken verglichen werden sollen. Wählen Sie dazu die notwendigen Spalten und bestätigen Sie die Eingabe mit der Plus-Taste.

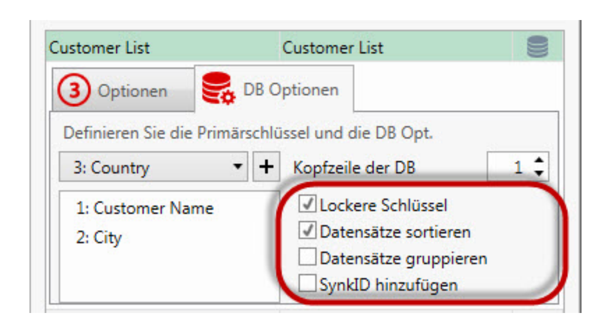


Wichtig:
Die Feldnamen/Spalten müssen identisch sein, da sonst die Datenbank kann nicht verglichen werden kann! Wenn keine identischen Feldnamen gefunden werden, erscheint die folgende Meldung:



In diesem Fall müssen Sie manuell in den Excel-Dateien die Feldnamen bzw. Spaltenüberschriften anpassen, damit Sie die Datenbanken vergleichen können.

5. Zum Schluss können Sie noch die Datenbankoptionen festlegen:



Die Datenbankoptionen bewirken folgendes:

Lockere Schlüssel

Im "normalen" Vergleich werden Primärschlüssel mit Unterschieden in Leerzeichen und Gross-/Kleinschreibung als Differenz wahrgenommen (Bsp: "Hans Meier" und "HANS MEIER"). Mit dieser Option werden Differenzen mit Leerzeichen und Gross-/Keinschreibung ignoriert.

Datensätze sortieren

Die Daten werden nach dem Primärschlüssel sortiert.

Datensätze gruppieren

Die Daten werden gruppiert. Die Datensätze werden wie folgt gruppiert:
1. verbundenen Datensätze
2. neue/gelöschte Datensätze
3. doppelten Primärschlüssel
4. doppelte Datensätze (redundante Datensätze)

SynkID hinzufügen

Rechts neben den Datenbanken wird eine neue Spalte mit der Bezeichnung "SynkID" hinzugefügt, die die Differenztypen enthält. Sie können dann die Datenbanken anhand der Differenztypen sortieren und/oder filtrieren (AutoFilter). Folgende Bezeichnungen werden in die SynkID-Spalte geschrieben:

0:OK	Unveränderte Zeile
1:DIFF	Zeile mit unterschiedlichen Zellwerten/Formeln
2:MISS	Nicht verbundene Zeile
3:DUPKEY	Doppelter Schlüssel
4:DUPREC	Duplikat (redundanter Datensatz)

6. Wenn Sie die Datenbankeinstellungen beendet haben, erscheint in der Tabellenliste neben dem Tabellenpaar ein Datenbanksymbol, welches signalisiert, dass die Dateien im Datenbankmodus verglichen werden.



2.2.3.1 Allgemeine Informationen über Datenbanken

Nicht alle Excel Anwender kennen den Unterschied zwischen "normalen" Tabellen und Datenbanken, darum wird nachfolgend kurz beschrieben, was deren Unterschiede sind.

"Normale" Tabelle

"Normale" Tabellen sind Arbeitsblätter, die keine Spaltenüberschriften und keine Datenbankstruktur enthalten. Synkronizer vergleicht hier jeden Zelleintrag miteinander. Der Datenvergleich von normalen Tabellen ist nicht so leistungsfähig wie bei Datenbanken. Die Daten müssen die gleiche Sortierreihenfolge aufweisen und auch das Löschen von Spalten/Zeilen innerhalb der Tabelle ist nicht empfehlenswert.

Datenbanken

Datenbanken sind Excel-Tabellen, die eine Datenbankstruktur aufweisen. Die Daten müssen wie folgt angeordnet sein:

- In der ersten Zeile der Tabelle müssen sich eindeutige Spaltenüberschriften befinden.

- Die Datenbank enthält eine Spalte mit dem **Primärschlüssel** (eindeutige Kennzahl, Artikelnummer usw.). Primärschlüssel werden benötigt, damit die Datensätze eindeutig identifiziert werden können. Die Definition eines Primärschlüssels ist nicht zwingend, jedoch sehr zu empfehlen. Fakultativ können Sie den Primärschlüssel auch aus mehreren Spalten bilden (z.B. Nachname, Vorname und Ort). Nachteil: es können doppelte Schlüssel oder Duplikate vorkommen.

Doppelte Schlüssel

Doppelte Schlüssel sind Datensätze, die die gleiche Identifikationsnummer bzw. Erkennung aufweisen. Angenommen, der Primärschlüssel wird aus dem Nach- und Vornamen gebildet und Sie haben mehrere "Hans Muster" in Ihrer Tabelle erfasst, dann weiss Synkronizer nicht, welche Datensätze miteinander verglichen werden sollen. **Achten Sie immer darauf, dass Sie mit**

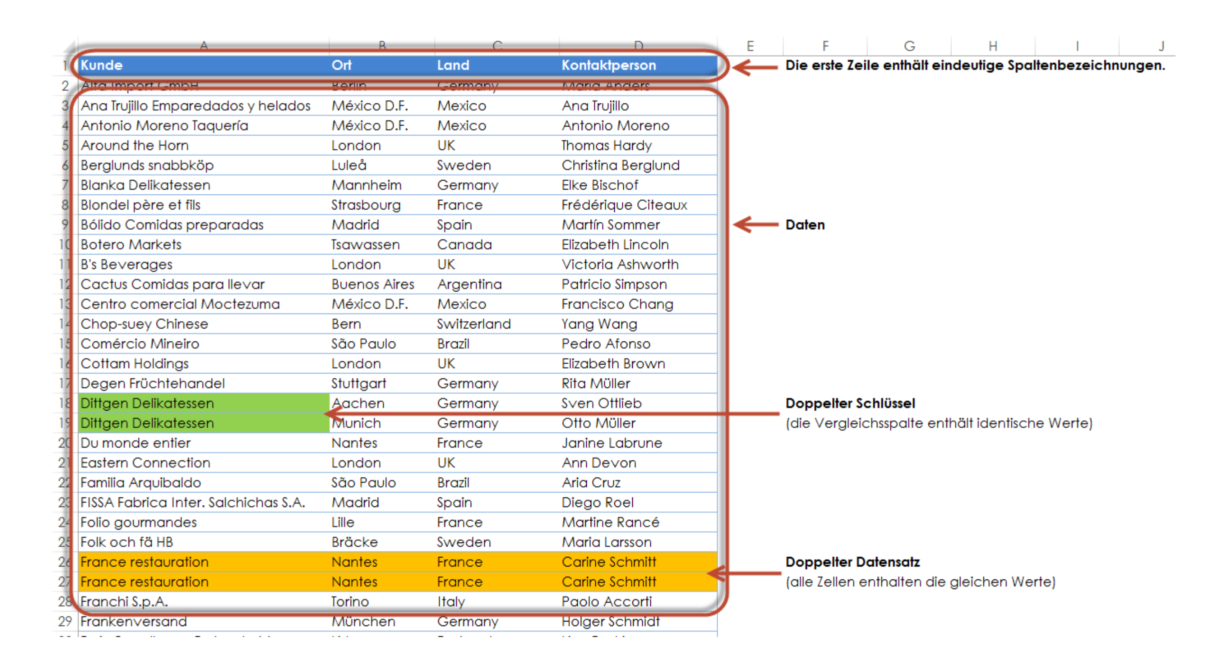
eindeutigen Schlüsseln arbeiten.

Doppelter Datensatz / Duplikat

Duplikate sind identische (redundante) Datensätze. Diese Datensätze sind überzählig und sollten gelöscht werden.

Der Vergleich von Datenbanken ist äusserst effizient. Die Datenbanken können beliebig sortiert und mutiert werden. Auch das Einfügen und Löschen von Spalten/Zeilen innerhalb der Tabellen ist erlaubt. Synkronizer erkennt hier sämtliche Abweichungen!

Schema Datenbank



	Kunde	Ort	Land	Kontaktperson
2	Alfa Import GmbH	Berlin	Germany	Maria Anders
3	Ana Trujillo Emparedados y helados	México D.F.	Mexico	Ana Trujillo
4	Antonio Moreno Taquería	México D.F.	Mexico	Antonio Moreno
5	Around the Horn	London	UK	Thomas Hardy
6	Berglunds snabbköp	Luleå	Sweden	Christina Berglund
7	Blanka Delikatessen	Mannheim	Germany	Elke Bischof
8	Blondel père et fils	Strasbourg	France	Frédérique Citeaux
9	Bólido Comidas preparadas	Madrid	Spain	Martín Sommer
10	Botero Markets	Tsawassen	Canada	Elizabeth Lincoln
11	B's Beverages	London	UK	Victoria Ashworth
12	Cactus Comidas para llevar	Buenos Aires	Argentina	Patricio Simpson
13	Centro comercial Moctezuma	México D.F.	Mexico	Francisco Chang
14	Chop-suey Chinese	Bern	Switzerland	Yang Wang
15	Comércio Mineiro	São Paulo	Brazil	Pedro Afonso
16	Cottam Holdings	London	UK	Elizabeth Brown
17	Degen Früchtehandel	Stuttgart	Germany	Rita Müller
18	Dittgen Delikatessen	Aachen	Germany	Sven Ottlieb
19	Dittgen Delikatessen	Munich	Germany	Otto Müller
20	Du monde entier	Nantes	France	Janine Labrune
21	Eastern Connection	London	UK	Ann Devon
22	Familia Arquibaldo	São Paulo	Brazil	Aria Cruz
23	FISSA Fabrica Inter. Salchichas S.A.	Madrid	Spain	Diego Roel
24	Folio gourmandes	Lille	France	Martine Rancé
25	Folk och fä HB	Bräcke	Sweden	Maria Larsson
26	France restauration	Nantes	France	Carine Schmitt
27	France restauration	Nantes	France	Carine Schmitt
28	Franchi S.p.A.	Torino	Italy	Paolo Accorti
29	Frankenversand	München	Germany	Holger Schmidt

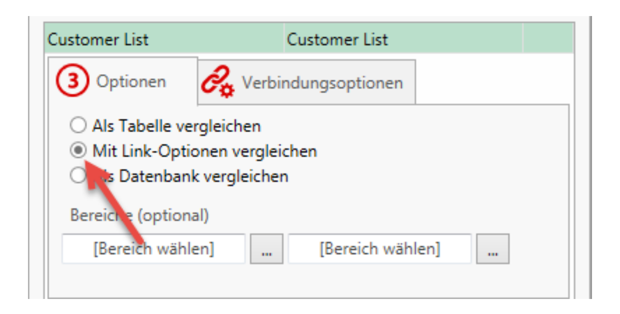
2.2.4 Daten 1:1 vergleichen

Wenn Sie sicher sind, dass Ihre Daten keine neuen oder gelöschten Zeilen/Spalten aufweisen, dann können Sie die Dateien auch mit der Methode 1:1 vergleichen. In diesem Fall werden die Zeilen bzw. Spalten 1:1 verglichen und es werden nur Zelldifferenzen erkannt.

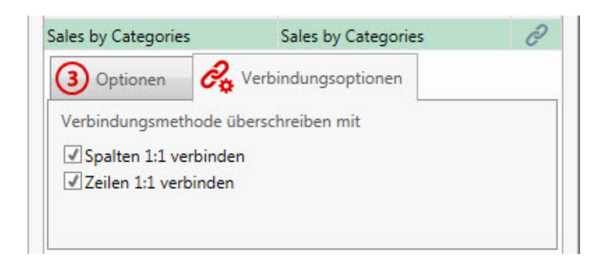
Gehen Sie wie folgt vor, um die Zeilen und/oder Spalten 1:1 zu vergleichen:

1. Wählen Sie das Tabellenpaar, welches 1:1 verglichen werden soll.

2. Klicken Sie auf "Als Tabelle vergleichen"



4. Das Register "Verbindungsoptionen" erscheint. Klicken Sie auf Zeilen und oder Spalten 1:1 verbinden.



5. Als Bestätigung der 1:1 Verbindungsmethode erscheint neben dem Tabellenpaar das Verbindungssymbol.

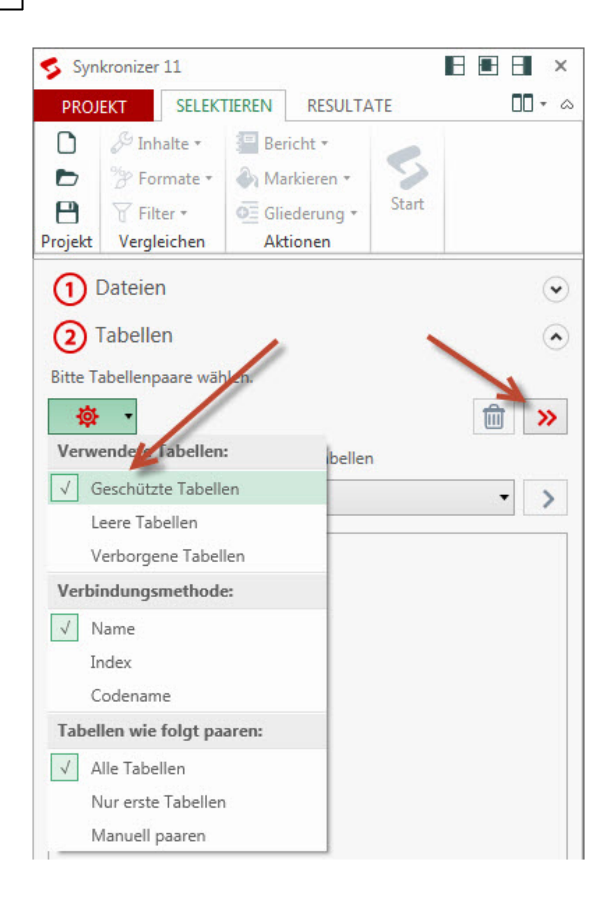


2.2.5 Kennwörter

Mit dem Synkronizer können auch kennwortgeschützte Tabellen verglichen werden. Der Nachteil ist aber, dass kennwortgeschützte Tabellen nicht aktualisiert und dass die Differenzen nicht markiert und/oder gegliedert werden können. Falls Sie aber das Kennwort/die Kennwörter der Tabelle(n) kennen, können Sie wie folgt die Tabellen entsperren:

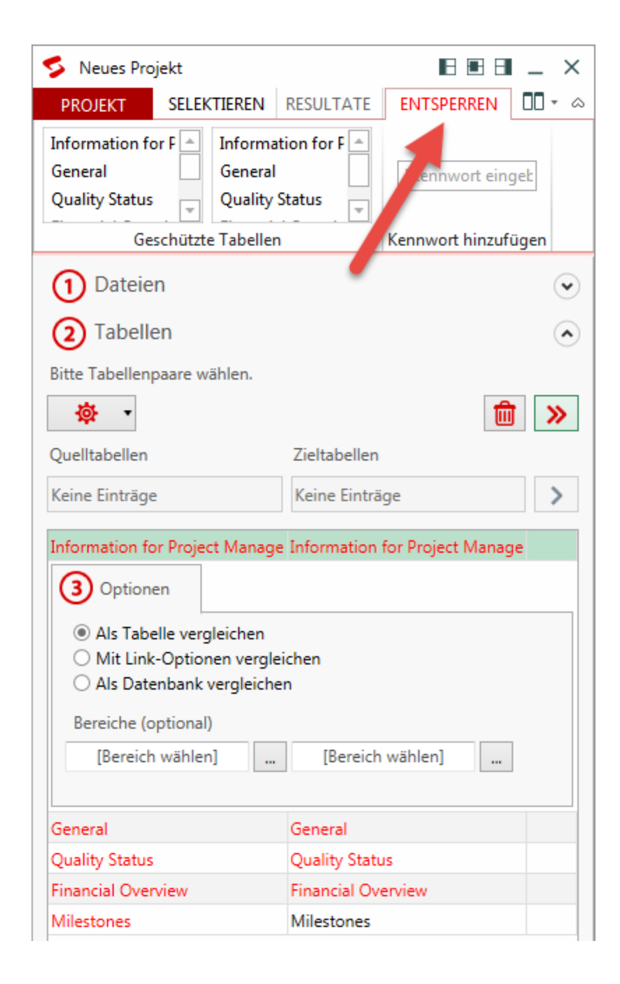
1. Öffnen Sie die Dateien, welche kennwortgeschütze Tabellen enthalten.

2. Wenn die Tabellen nicht in der Tabellenliste erscheinen sollten, müssen Sie die "geschützten Tabellen" im Tabellenpaarungs-Element aktivieren:

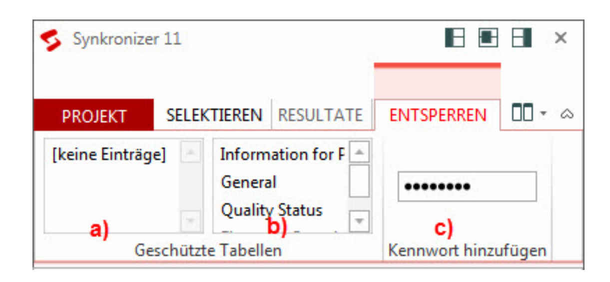


Selektieren Sie "Geschützte Tabellen" und klicken auf die "Tabellen hinzufügen"-Taste - die geschützten Tabellen erscheinen nun.

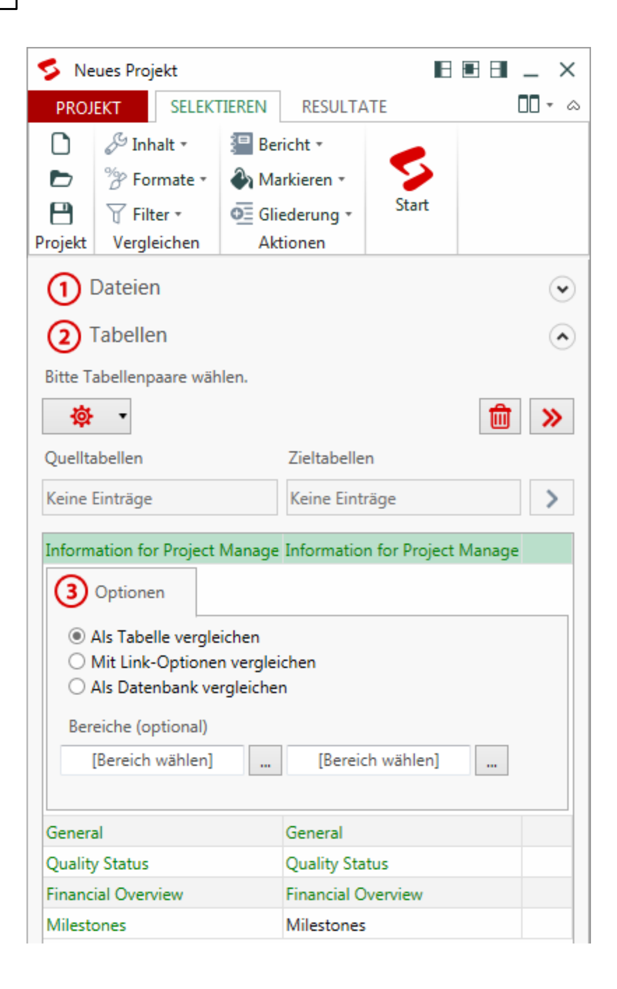
3. Die kennwortgeschützten Tabellen werden rot oder grün angezeigt (rot bedeutet, dass die Tabelle geschützt ist, grün bedeutet, dass die Tabelle erfolgreich entsperrt werden konnte). Zusätzlich erscheint ein Register "Entsperren", in welchem Sie die Kennwörter zum Entsperren der Tabellen eingeben können:



4. Im Register "Entsperren" sind drei Spalten ersichtlich. In der ersten Spalte (a) sind sämtliche Tabellen der ersten Datei gelistet, welche noch kennwortgeschützt sind, in der zweiten Spalte (b) sind die kennwortgeschützten Tabellen der zweiten Datei gelistet. In der dritten Spalte (c) werden die Kennwörter eingegegen, um die Tabellen zu entsperren.



5. Geben Sie die Kennwörter im Feld c ein und entsperren die Tabellen. Wenn Sie alle Kennwörter erfolgreich eingegeben haben, werden alle Tabellen grün dargestellt und das Register "Entsperren" verschwindet.



Kennwörter in Projekten

Die Tabellen-Kennwörter werden nicht in den Projekten gespeichert.

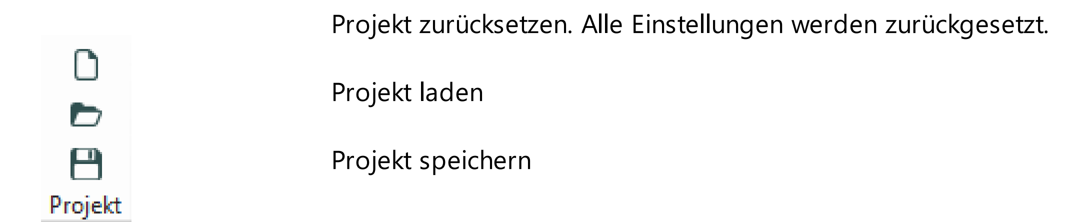
2.3 Symbolleiste

Im oberen Teil des Selektieren-Registers sind verschiedene Befehle verfügbar. Sie können Projekte verwalten, Vergleichseinstellungen definieren und den Tabellenvergleich starten.



Die folgenden Befehle sind verfpgbar:

Projekt Befehle

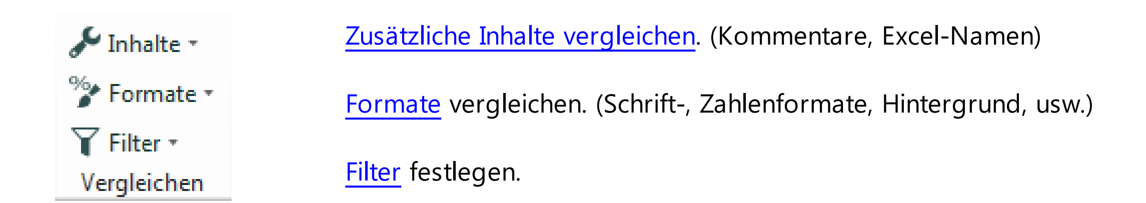


Projekt zurücksetzen. Alle Einstellungen werden zurückgesetzt.

Projekt laden

Projekt speichern

Vergleichs-Befehle

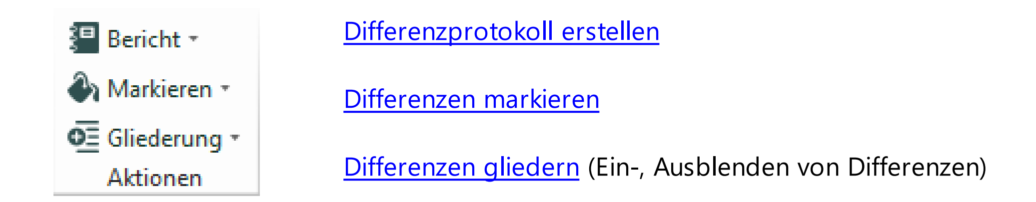


Zusätzliche Inhalte vergleichen. (Kommentare, Excel-Namen)

Formate vergleichen. (Schrift-, Zahlenformate, Hintergrund, usw.)

Filter festlegen.

Aktions-Befehle

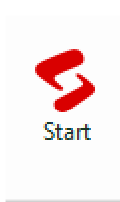


Differenzprotokoll erstellen

Differenzen markieren

Differenzen gliedern (Ein-, Ausblenden von Differenzen)

Start



Der Tabellenvergleich wird gestartet.

2.3.1 Projekt

Mit den "Projekt"-Befehlen haben Sie die Möglichkeit, stets wiederkehrende Vergleichsprozesse als Projekt zu speichern. Angenommen, Sie müssen immer die gleichen Dateien miteinander vergleichen, dann können Sie diese(n) Vergleichsprozess(e) als Projekt speichern. Das nächste Mal öffnen Sie einfach dieses Projekt und Sie können sofort mit dem Vergleich beginnen.



Folgende Befehle stehen zur Auswahl:

Neu

Das Formular wird geleert, wobei sämtliche Einstellungen zurückgestellt werden

Öffnen

Mit diesem Befehl wird ein bestehendes Projekt geöffnet

Speichern

Die aktuellen Einstellungen werden als Projekt gespeichert

Im Projekt-Register können ebenfalls die Projekte verwaltet werden.

Kennwörter in Projekten

Die Tabellen-Kennwörter werden nicht in den Projekten gespeichert.

2.3.2 Vergleichen - Inhalte

In diesem Menü können Sie bestimmen, ob zusätzliche Inhalte verglichen werden sollen. Folgende Inhalte können gewählt werden:

- **Kommentare**
  (Zell-)Kommentare sind Informationen, die jeder Zelle hinterlegt werden können.

- **Namen**
  Namen sind Information, welche für einen Zellbereich, eine Funktion, eine Konstante oder eine Tabelle festlegt werden können.

Gehen Sie wie folgt vor, um einen Inhalt zusätzlich zu vergleichen.

1. Klicken Sie auf Inhalt.

2. Aktivieren Sie den/die gewünschte(n) Inhalt(e).

3. Das Inhalts-Symbol ändert von grau auf schwarz.

Anmerkung:
Es werden in zukünftigen Synkronizer Versionen weitere Inhalte verglichen werden können.

2.3.3 Vergleichen - Formate

Folgende Zellformate können verglichen werden:

- **Zahlen**
  Es wird verglichen, ob die Zahlenformatierung der Zellen unterschiedlich ist.

- **Ausrichtung**
  Es wird kontrolliert, ob die Ausrichtung des Textes unterschiedlich ist (linksbündig, zentriert, usw.)

- **Schrift**
  Sämtliche Schriftformatierungen werden miteinander verglichen (Fontname, Grösse, Farbe, Unterstreichung, usw.)

- **Rahmen**
  Die Zellrahmen werden verglichen.

- **Ausfüllen**
  Die Hintergrundfarbe der Zellen wird miteinander verglichen

- **Schutz**
  Der Zellschutz und die Sichtbarkeit der Zelle werden verglichen.

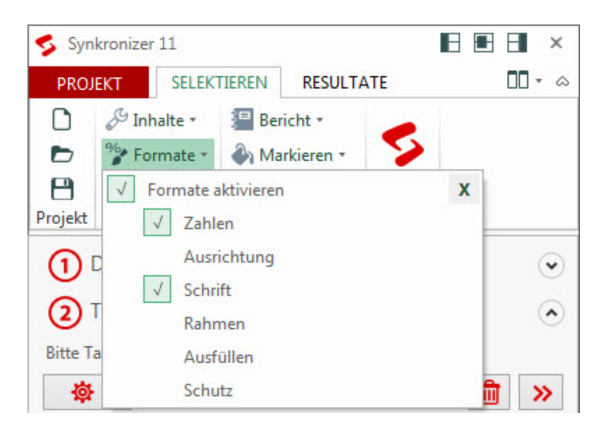
Gehen Sie wie folgt vor, um die Zellformate zu vergleichen:

1. Klicken Sie auf Formate

2. Wählen Sie "Formate aktivieren" (Das Formate-Symbol ändert von grau auf schwarz).

3. Selektieren Sie alle Formate, die Sie vergleichen möchten.

In diesem Beispiel werden die Zahlen- und Schriftformate verglichen.



2.3.4 Vergleichen - Filter

Mit den Filterfunktionen haben Sie die Möglichkeit, die Anzahl der Differenzen zu verringern. Klicken Sie einfach auf "Filter aktivieren" und die Filter, die Sie anwenden möchten.

- **Gross-/Keinschreibung ignorieren**
  Wenn dieses Feld aktiviert ist, werden Unterschiede in der Gross-/Kleinschreibung ignoriert (z.B. "Hans Muster", "HANS MUSTER").

- **Leerzeichen ignorieren**
  Falls dieses Kästchen aktiviert ist, werden Leerzeichen zu Beginn und am Ende der Wörter ignoriert, z.B. die Werte "Hans Muster" und " Hans Muster " (Leerzeichen zu Beginn und am Schluss) werden nicht als Differenz wahrgenommen.

- **Datentyp ignorieren**

Es kann vorkommen, dass Zahlen in der einen Tabelle als Zahl, in der anderen aber als Text formatiert sind. Dieser Unterschied führt logischerweise zu einer Differenz, obwohl die Zahlen identisch sind. Mit dieser Option werden diese Differenzen ignoriert.

- **Formeln mit identischem Wert ign.**
  Angenommen, Sie haben zwei unterschiedliche Formeln, die beide das gleiche Ergebnis ausweisen. Siehe das folgende Beispiel:

  Zelle Quelldatei
  Formel: =D50
  Ergebnis: 100

  Zelle Zieldatei
  Formel: =$D$50
  Ergebnis: 100

  Diese Art von Differenzen werden mit dieser Option rausgefiltert.

- **Konstanten ignorieren**
  Ist diese Optionen aktiv, werden sämtliche Zellen mit Konstanten (eingegebenen Werten) ignoriert.

- **Formeln ignorieren**
  Hier werden alle Formelfelder rausgefiltert.

- **Verborgene Spalten ignorieren**
  Alle Spalten, die in der Quelldatei ausgeblendet sind, werden ignoriert. Ausgeblendete Spalten der Zieldatei werden hingegen berücksichtigt.

- **Verborgene Zeilen ignorieren**
  Funktioniert gleich wie "verborgene Spalten", jedoch mit Zeilen.

- **Numerische Toleranz**
  Bei numerischen Feldern können Sie bestimmen, wie der Synkronizer mit Differenzen umgehen soll. Angenommen es sollen nur Differenzen, die grösser als 0.10 sind, ausgewiesen werden, dann geben Sie hier 0.10 ein. Die Differenzen bis und mit 0.10 werden dann ignoriert.

- **Gleichwertige Werte**
  Bei gewissen Tabellen kommt es vor, dass die selbe Differenz massenhaft auftaucht. Angenommen Sie vergleichen Checklisten, die eine Erledigt-Spalte haben. In der einen Liste werden in der Erledigt-Spalte die Werte "ja/nein" eingegeben, in der anderen aber "yes/no" oder "1/0". Bei einem normalen Vergleich würden etliche Differenzen ausgspukt. Sie können nun diese Differenzen rausfiltern, indem Sie die entsprechenden Äquivalente eingeben, z.B. 1=yes, 1=ja, yes=ja, 0=no, 0=nein, no=nein usw.

Gehen Sie wie folgt vor, um einen oder mehrere Filter zu aktivieren:

1. Klicken Sie auf Filter

2. Wählen Sie "Filter aktivieren". (Das Filter-Symbol ändert von grau auf schwarz)

3. Selektieren Sie die Filter, die Sie aktivieren möchten.

4. Schliessen Sie das Filter-Fenster.

2.3.5 Aktionen - Bericht

Im Menü Bericht können Sie bestimmen, ob beim Vergleichen der Seite zusätzlich ein Differenzprotokoll erstellt werden soll oder nicht. Folgende Optionen stehen zur Auswahl:

- **Kein Differenzprotokoll**
  Es wird kein Differenzprotokoll erstellt.

- **Differenzprotokoll**
  Es wird ein "normales" Differenzprotokoll erstellt.

- **Differenzprotokoll mit Hyperlinks**
  Es wird ein Differenzprotokoll mit Hyperlinks erstellt.

Das Berichts-Symbol wird je nach Status entweder grau oder schwarz dargestellt. Wenn ein Differenzprotokoll erzeugt werden soll, ist das Symbol schwarz, sonst grau.

2.3.6 Aktionen - Markieren

In diesem Menü bestimmen Sie, ob die Differenzen in den zugrundeliegenden Exceltabellen markiert werden sollen oder nicht. Folgende Optionen stehen zur Verfügung:

- **Nicht markieren**
  Die Exceltabellen bleiben unangetastet.

- **Differenzen markieren**
  Alle Differenzen werden in den zugrundeliegenden Exceltabellen markiert.

- **Differenzen löschen und markieren**
  Hier werden ebenfalls alle Differenzen markiert. Zusätzlich werden vor dem Vergleich sämtliche Hintergrundfarben der Exceltabellen (Füllfarbe der Zellen) gelöscht.

Je nach Status wird das Markierungs-Symbol grau oder schwarz dargestellt. Wenn die Differenzen markiert werden sollen, ist das Symbol schwarz, sonst grau.

2.3.7 Aktionen - Gliederung

Mit dem Gliederungs-Menü können Sie entscheiden, ob Sie nur bestimmte Differenztypen sehen möchten oder nicht. Auf diese Weise können Sie die Differenzen sehr effizient prüfen und abarbeiten. Folgende Differenztypen können gewählt werden:

- **Nicht verbundene Spalten**
  Diese Spalten kommen nur in einer Tabelle vor.

- **Nicht verbundene Zeilen**
  Diese Zeilen wurden entweder gelöscht oder hinzugefügt.

- **Doppelte Zeilen**
  Betrifft nur Datenbankmodus: diese Zeilen beinhalten entweder doppelte (redundante) Datensätze oder Zeilen mit doppelten Schlüsseln.

- **Identische Zeilen**
  Diese Zeilen sind komplett identisch und weisen keine Zelldifferenzen aus.

- **Unterschiedliche Zeilen**
  Diese Zeilen enthalten Zelldifferenzen.

Gehen Sie wie folgt vor, wenn nur bestimmte Differenzen angezeigt werden sollen:

1. Klicken Sie auf Gliederung

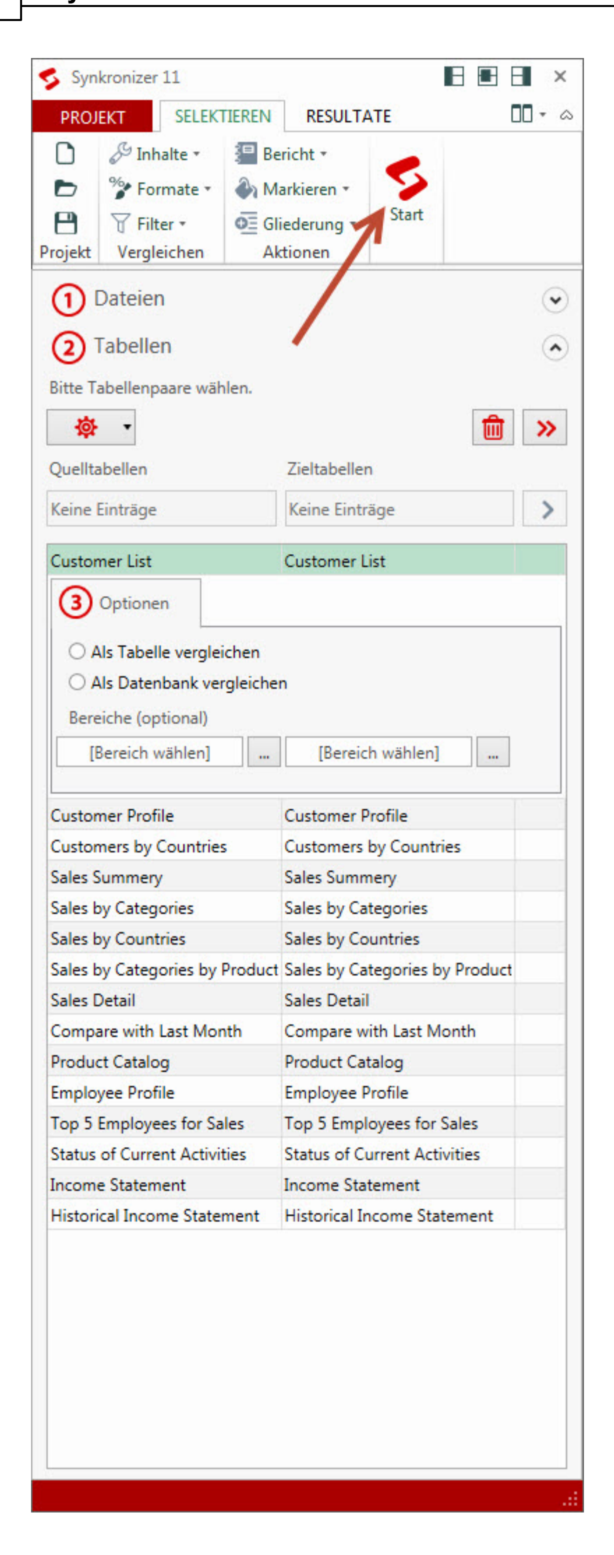
2. Wählen Sie "Gliederung aktivieren" (Das Gliederungs-Symbol ändert von grau auf schwarz).

3. Selektieren Sie die Differenztypen, die Sie anzeigen möchten.

4. Schliessen Sie das Gliederungs-Fenster.

2.3.8 Start - Dateien vergleichen

Wenn Sie sämtliche Tabellen, Eigenschaften und Vergleichsoptionen festgelegt haben, können Sie den Datenvergleich starten. Klicken Sie einfach auf das Start-Symbol, worauf die Dateien verglichen werden.

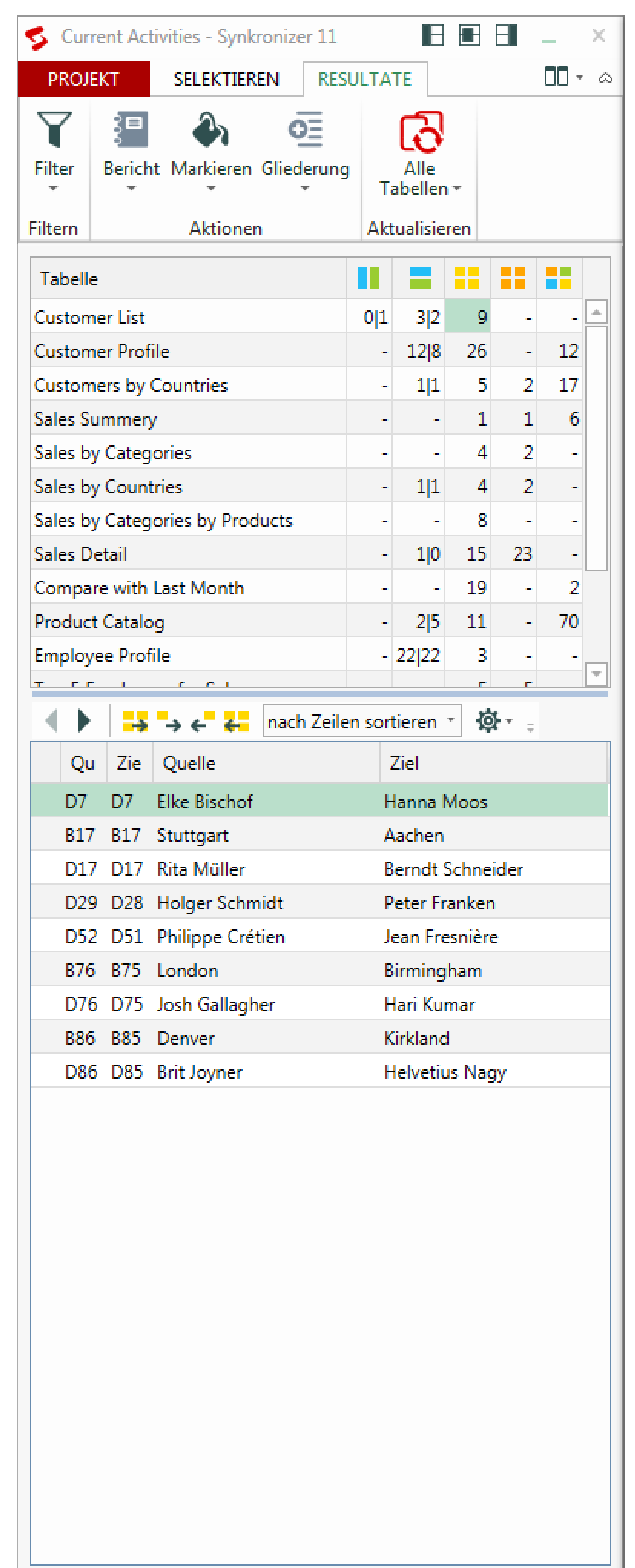


Nach erfolgtem Vergleich werden die Dateien und das Resultate-Register nebeneinander dargestellt. Mit dem Resultate-Register können Sie dann die Differenzen elegant kontrollieren und abarbeiten.

Part III

3 Resultate Register

Nach dem Vergleich werden das Resultateformular und die Tabellen nebeneinander dargestellt. Mit dem Resultateformular können Sie dann elegant jede Differenz prüfen und abarbeiten.



Symbolleiste

In diesem Bereich können verschiedene Aktionen durchgeführt werden. Mit diesen Schaltflächen können Sie gezielt und effizient die Differenzen überprüfen und verwalten.

Differenzübersicht

In diesem Bereich werden die Arbeitsblätter und die Anzahl der gefundenen Differenzen gezeigt. Die Detailinformationen für das ausgewählte Arbeitsblatt/Kategorie werden im unteren Fensterbereich angezeigt.

Detailinformationen der Differenzen

Dieser Bereich zeigt jeden Unterschied der oben gewählten Arbeitsblatt/ Kategorien-Zelle. Sie können jede Differenz prüfen und entscheiden, ob die Differenz übertragen werden muss oder nicht.

Der Synkronizer sorgt automatisch dafür, dass die richtigen Ausschnitte in beiden Tabellenfenstern zu sehen sind, dass die fraglichen Stellen direkt miteinander verglichen werden können. Der Anwender entscheidet dann, mit welcher Version bzw. Excel-Datei in Zukunft gearbeitet werden soll. Per Mausklick kann man den Wert aus der Quelldatei in die Zieldatei oder umgekehrt übernehmen. Auf diese Weise können die Unterschiede in

kürzester Zeit abgearbeitet werden. Nach wenigen Mausklicks verfügen man über eine aktualisierte Excel-Tabelle, mit der wie gewohnt weiter gearbeitet werden kann.

Aktionen

Mit dem Resultateformular lassen sich folgende Aktionen durchführen:

- Übersicht der Differenzen
- Arbeitsblätter abgleichen / Differenzen transferieren
- Unerwünschte Differenzen löschen
- Gewünschte Differenzen ein- und ausblenden

3.1 Übersicht

Im oberen Teil des Resultateformulars werden die Tabellen und deren Differenzen angezeigt. Auf einen Blick sehen Sie, wie viele Differenzen in den Tabellen gefunden wurden.

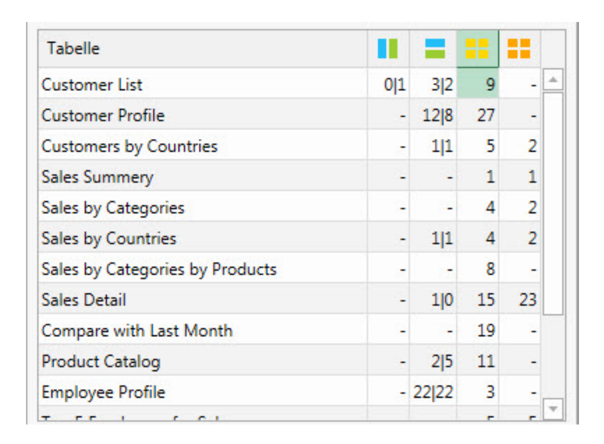


Tabelle				
Customer List	0|1	3|2	9	-
Customer Profile	-	12|8	27	-
Customers by Countries	-	1|1	5	2
Sales Summery	-	-	1	1
Sales by Categories	-	-	4	2
Sales by Countries	-	1|1	4	2
Sales by Categories by Products	-	-	8	-
Sales Detail	-	1|0	15	23
Compare with Last Month	-	-	19	-
Product Catalog	-	2|5	11	-
Employee Profile	-	22|22	3	-

In den Spalten des Listenfeldes werden die verschiedenen Differenzen angezeigt. In diesem Screenshot wurden z.B. in der Tabelle "Customer List" die folgenden Differenzen gefunden:

Nicht verbundene Spalten	0|1; (keine Spaltendifferenz in Quelltabelle, eine Spaltendifferenz in Zieltabelle)
Nicht verbundene Zeilen	3|2; (drei Zeilendifferenzen in Quelltabelle, zwei Zeilendifferenzen in Zieltabelle)
Unterschiede in eingegebenen Zellwerten/-formeln	9 Differenzen in eingegebenen Zellen und/oder Formeln
Unterschiede in kalkulierten Werten	- (keine Unterschiede)

Folgende Differenztypen werden ausgewiesen:

Nicht verbundene Spalten. Diese Spalten sind nur in einer Tabelle vorhanden.

Doppelte Zeilen (redundante Zeilen). Bei diesen Datensätzen ist jede Zelle identisch. Diese Zeilen sollten gelöscht werden (erscheint nur beim Datenbankvergleich).

Doppelte Schlüssel. Bei diesen Datensätzen ist der selbe Primärschlüssel mehrmals vorhanden (erscheint nur beim Datenbankvergleich).

Nicht verbundene Zeilen. Diese Zeilen sind nur in einer Tabelle vorhanden.

Unterschiede in eingegebenen Zellwerten und/oder -formeln. Bei diesen Differenzen ist der eingegebene Zellwert oder die eingegebene Formel unterschiedlich.

Unterschiede in kalkulierten Werten. Bei diesen Differenzen ist das kalkulierte Ergebnis unterschiedlich (die eingegebene Formel ist identisch).

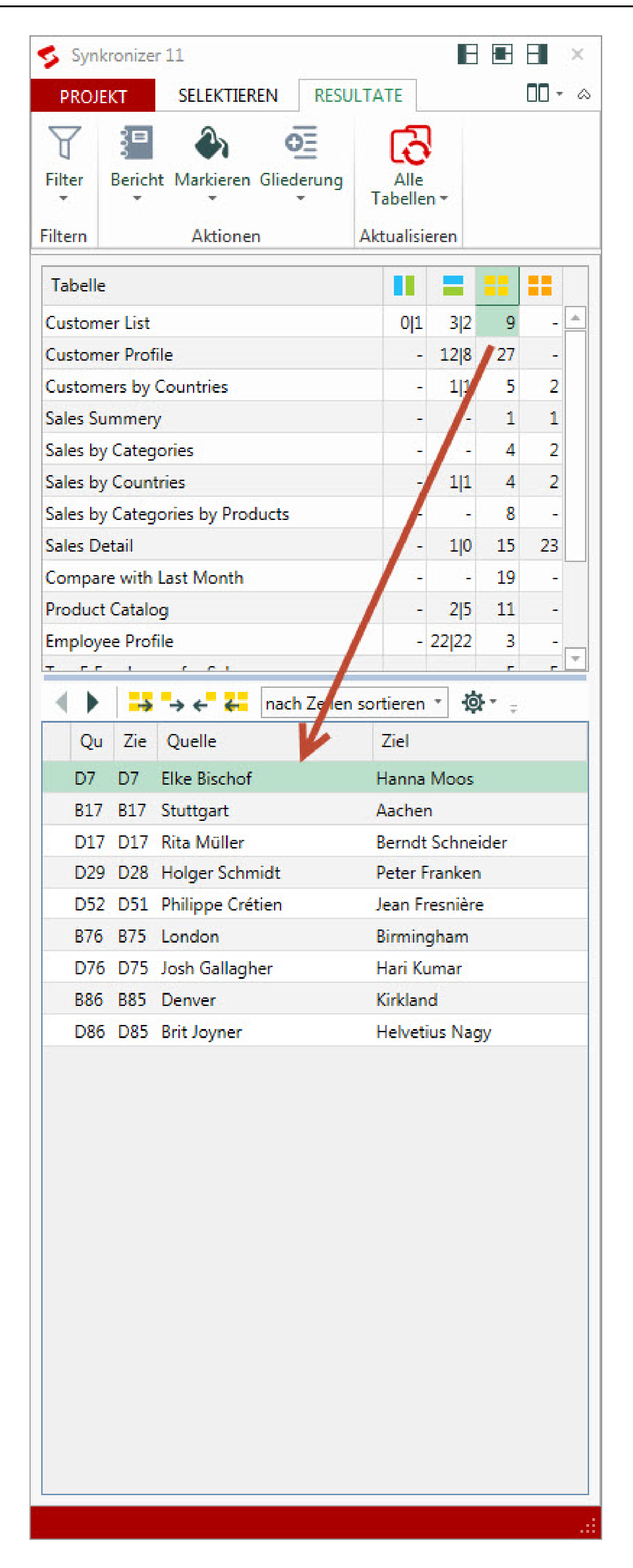
Unterschiedliche Zellformate (erscheint nur beim Formatevergleich).

Unterschiedliche Kommentare (erscheint nur beim Kommentarevergleich).

Unterschiedliche Namen (erscheint nur beim Namensvergleich)

Für jede Differenz können die Detailinformationen angezeigt werden, die dann im unteren Teil des Formulares erscheinen. Klicken Sie einfach mit der Maus auf den gewünschten Differenztyp.

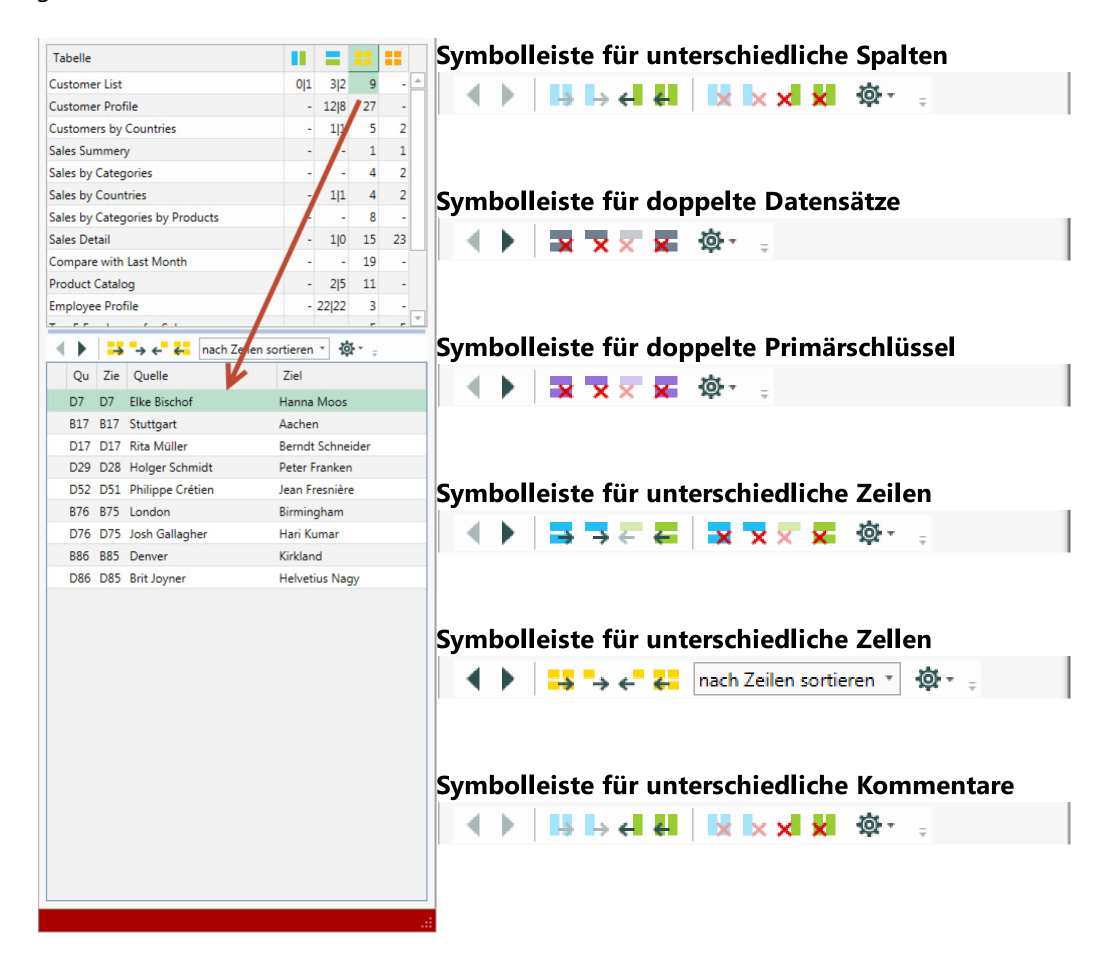
Im folgenden Beispiel wurde der Differenztyp "Eingegebene Werte & Formeln" gewählt. Es werden im oberen Teil des Formulars 9 Differenzen angezeigt, die im unteren Teil einzeln ausgewiesen werden.



Synkronizer 11
PROJEKT
SELEKTIEREN
RESULTATE
Filter
Bericht
Markieren
Gliederung
Alle Tabellen
Filtern
Aktionen
Aktualisieren
Tabelle
Customer List 0|1 3|2 9 -
Customer Profile - 12|8 27 -
Customers by Countries - 1|1 5 2
Sales Summery - - 1 1
Sales by Categories - - 4 2
Sales by Countries - 1|1 4 2
Sales by Categories by Products - - 8 -
Sales Detail - 1|0 15 23
Compare with Last Month - - 19 -
Product Catalog - 2|5 11 -
Employee Profile - 22|22 3 -
nach Zeilen sortieren
Qu Zie Quelle Ziel
D7 D7 Elke Bischof Hanna Moos
B17 B17 Stuttgart Aachen
D17 D17 Rita Müller Berndt Schneider
D29 D28 Holger Schmidt Peter Franken
D52 D51 Philippe Crétien Jean Fresnière
B76 B75 London Birmingham
D76 D75 Josh Gallagher Hari Kumar
B86 B85 Denver Kirkland
D86 D85 Brit Joyner Helvetius Nagy

3.2 Detailinformationen

Im unteren Teil des Formulars werden die Detailinformationen angezeigt, die im oberen Teil gewählt wurden



Sämtliche Differenzen können abgearbeitet, transferiert oder gelöscht werden, wofür folgende Schaltflächen zur Verfügung stehen:



Differenzen wählen

Mit diesen Schaltflächen kann man zur nächsten bzw. vorherigen Differenz springen.





Unterschiedliche Spalten/Zeilen von Quelldatei nach Zieldatei transferieren

Mit diesen Schaltflächen werden die Spalten/Zeilen von der Quell- in die Zieldatei übertragen (von links nach rechts). Es werden entweder alle oder nur die markierte Spalte/Zeile transferiert. Sobald die Spalte/Zeile transferiert wurde, wird sie rot dargestellt.

Unterschiedliche Spalten/Zeilen von Zieldatei nach Quelldatei transferieren

Wie oben, nur in die andere Richtung.

Unterschiedliche Spalten/Zeilen in Quelldatei löschen

Mit diesen Schaltflächen werden die Spalten/Zeilen in der Quelldatei (links) gelöscht. Es werden entweder alle oder nur die markierte Spalte/Zeile gelöscht.

Unterschiedliche Spalten/Zeilen in Zieldatei löschen

Wie oben

Zelldifferenzen/-kommentare von Quelldatei nach Zieldatei transferieren

Mit diesen Schaltflächen werden die unterschiedlichen Zellwerte oder -kommentare von der Quell- in die Zieldatei übertragen (von links nach rechts). Es werden entweder alle oder nur die markierte Differenz transferiert. Sobald die Differenz transferiert wurde, wird sie rot dargestellt.

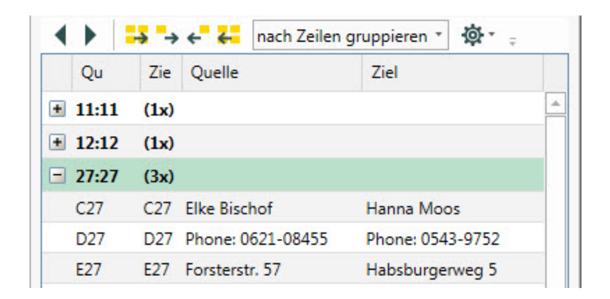
Zelldifferenzen/-kommentare von Zieldatei nach Quelldatei transferieren

Wie oben

nach Zeilen sortieren Mit dieser Schaltfläche kann die Sortierung der Zelldifferenzen geändert werden. Je nach Bedarf können die Zelldifferenzen anders sortiert oder gruppiert werden. Die Zelldifferenzen werden standardmässig nach Zeilen und innerhalb der Zeilen nach Spalten sortiert.

nach Spalten sortieren Die Zelldifferenzen werden zuerst nach Spalten und innerhalb der Spalten nach der Zeilennummer sortiert.

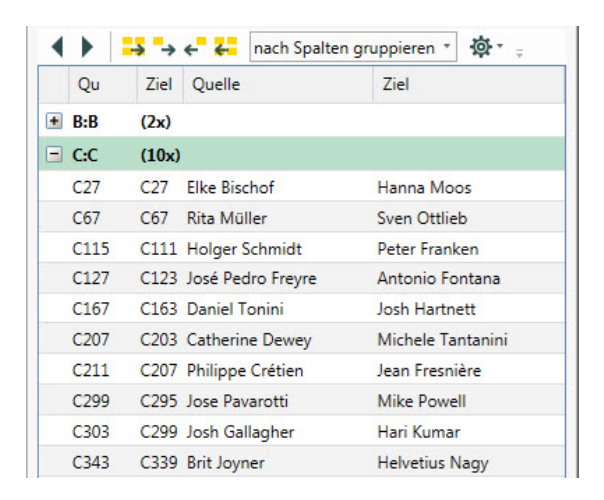
nach Zeilen gruppieren Wenn man diese Option wählt, werden die Zelldifferenzen nach Zeilen gruppiert. Die Differenzen können je nach Bedarf einzeln oder gruppenweise pro Zeile übertragen werden.



Qu	Zie	Quelle	Ziel
11:11	**(1x)**		
12:12	**(1x)**		
27:27	**(3x)**		
C27	C27	Elke Bischof	Hanna Moos
D27	D27	Phone: 0621-08455	Phone: 0543-9752
E27	E27	Forsterstr. 57	Habsburgerweg 5

nach Spalten gruppieren

Wenn man diese Option wählt, werden die Zelldifferenzen nach Spalten gruppiert. Die Differenzen können je nach Bedarf einzeln oder gruppenweise pro Spalte übertragen werden.



Qu	Ziel	Quelle	Ziel
B:B	**(2x)**		
C:C	**(10x)**		
C27	C27	Elke Bischof	Hanna Moos
C67	C67	Rita Müller	Sven Ottlieb
C115	C111	Holger Schmidt	Peter Franken
C127	C123	José Pedro Freyre	Antonio Fontana
C167	C163	Daniel Tonini	Josh Hartnett
C207	C203	Catherine Dewey	Michele Tantanini
C211	C207	Philippe Crétien	Jean Fresnière
C299	C295	Jose Pavarotti	Mike Powell
C303	C299	Josh Gallagher	Hari Kumar
C343	C339	Brit Joyner	Helvetius Nagy

Diese Option ist enorm praktisch, wenn Sie z.B. nur eine Spalte einer Datenbank vergleichen bzw. abgleichen möchten. **Mit nur einem Klick können alle Differenzen einer Spalte übertragen werden.**

Einstellungen

Je nach Differenztyp können verschiedene Optionen eingestellt werden, die nachfolgend beschrieben werden.

- **Ganze Spalten/Zeilen einfügen**
  Diese Option ist nur wichtig, wenn Sie mit Zellbereichen arbeiten. Normalerweise werden bei einer Spalten-/Zeileneinfügung auch die Spalten/Zeilen ausserhalb des Zellbereichs hinzugefügt. Durch Deaktivierung dieser Option werden nur noch Spalten/Zeilen innerhalb des Zellbereichs eingefügt.

  Diese Option betrifft nur die Kategorien Spalten und Zeilen.

- **Leere Zellen dürfen Daten überschreiben**
  Wenn dieses Feld markiert ist, werden Zellen mit Werten durch leere Zellen überschrieben. Falls dies nicht erwünscht ist, können Sie das entsprechende

Kontrollkästchen deaktivieren.

Differenzen, die mit einem Affenschwanzzeichen (@) dargestellt werden

Wenn eine Zahl, ein Datum oder eine boolesche Variable (Wahr/Falsch) als Text formatiert wurden, werden diese Differenzen mit einem Affenschwanzzeichen (@) dargestellt. Z.B. die Zahl 123 als Text formatiert wird so dargestellt: @:123.

3.3 Symbolleiste

Zusätzlich können im Resultateformular verschiedene Befehle ausgeführt werden, mit denen Sie die Differenzen gezielt und effizient verwalten können.



Filter

Bei Bedarf können Sie nach dem Vergleich einen anderen Filter wählen und die Daten neu vergleichen. Auf diese Weise sehen Sie sofort, ob Sie bessere Vergleichsergebnisse erzielen oder nicht. Gehen Sie wie folgt vor, um eine neue Filterabfrage zu starten:

1. Klicken Sie auf den Pfeil unter dem Filter-Symbol.
2. Wählen Sie Filter aktivieren (das Filter Symbol ändert sich)
3. Klicken Sie auf alle Filter, die Sie aktivieren oder deaktivieren möchten.
4. Schliessen Sie das Filterfenster.
5. Klicken Sie auf das "Aktualisieren"-Symbol.
6. Die Tabelle(n) wird/werden neu verglichen.

Genauere Informationen dazu entnehmen Sie dem Kapitel Thema.



Bericht

Sie können jederzeit ein Abweichungsprotokoll der Differenzen erzeugen. Gehen Sie wie folgt vor:

1. Klicken Sie auf den Pfeil unter dem Bericht-Symbol.
2. Wählen Sie das gewünschte Abweichungsprotokoll, das Sie erstellen wollen.
3. Das Abweichungsprotokoll wird erzeugt.



Markieren

Hier können Sie entscheiden, ob die Differenzen in den Excel-Tabellen markiert werden sollen oder nicht. Bei Bedarf können Sie die Markierungen wieder entfernen. Wählen Sie einfach "keine Markierung / Markierung entfernen".

Gliederung

Hier können Sie entscheiden, ob nur bestimmte Differenztypen gezeigt werden sollen. Gehen Sie wie folgt vor, um nur bestimmte Differenztypen anzuzeigen:

1. Klicken Sie auf den Pfeil unter dem "Gliederungs"-Symbol.
2. Wählen Sie Gliederung aktivieren (das "Gliederungs"-Symbol ändert sich)
3. Klicken auf die Differenztypen, die angezeigt werden sollen.
4. Schliessen Sie das Gliederungsfenster.
5. Klicken Sie auf das "Gliederungs"-Symbol.
6. Es werden nur noch die gewünschten Differenztypen angezeigt.



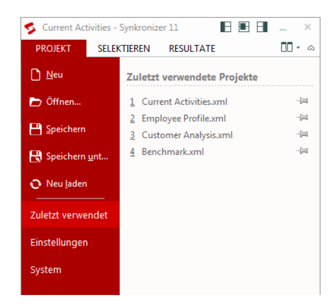
Aktualisieren

Beim Klick auf dieses Symbol werden die Dateien neu verglichen. Sie können wählen, ob alle Tabellen oder nur die aktive Tabelle neu verglichen werden soll.

Part IV

4 Projekt Register

In diesem Register können Sie Projekte verwalten, Projekteinstellungen tätigen und Informationen über Ihre Synkronizer Edition entnehmen.



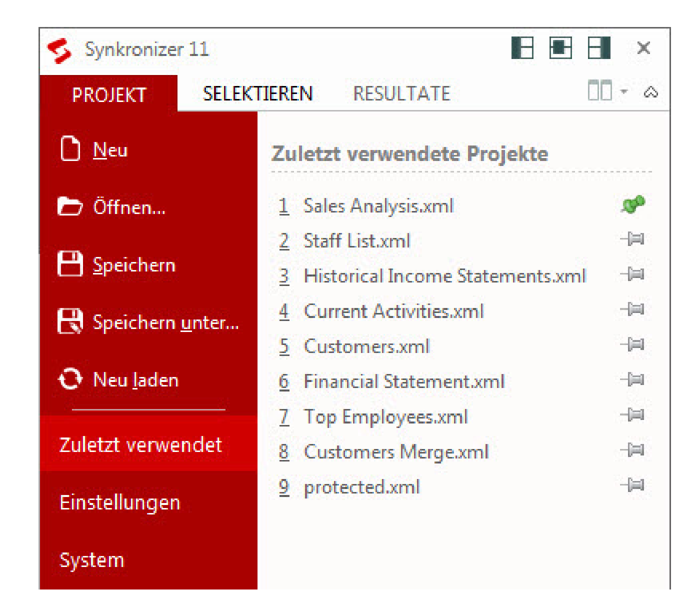
Die folgenden Information/Aktionen können in diesem Register entnehmen bzw. festlegen:

Projekte verwalten

Globale Einstellungen festlegen

System-/Lizenzinformationen

4.1 Projekte



Sämtliche Vergleichsprozesse können in Projekten gespeichert werden. Das hat den grossen Vorteil, dass Sie nicht jedes Mal die Dateien, Tabellen und Vergleichsinformationen neu eingeben müssen.

In diesem Menü stehen Ihnen folgende Befehle zur Verwaltung der Projekte zur Verfügung:

Neu

Das Formular wird geleert, wobei sämtliche Einstellungen zurückgestellt werden.

Öffnen

Mit diesem Befehl wird ein bestehendes Projekt geöffnet.

Speichern

Die aktuellen Einstellungen werden als Projekt gespeichert.

Speichern unter...
Das gewählte Projekt wird unter einem neuen Namen gespeichert.

Neu laden
Das Projekt wird neu geladen, wobei alle alten Einstellungen verlorengehen.

Zuletzt verwendet / Zuletzt verwendete Projekte
Liste der zuletzt verwendeten Projekte. Das Projekt wird neu geladen, wobei alle alten Einstellungen verlorengehen.

Die meisten Projekte-Befehle sind auch im Selektieren-Register verfügbar.

Kennwörter in Projekten

Die Tabellen-Kennwörter werden nicht in den Projekten gespeichert.

4.1.1 Definition

Die Projektdatei ist eine einfache XML Textdatei, die mit einem Texteditor (z.B. Notepad) bearbeitet werden kann. Wenn Sie manuell XML Dateien editieren, müssen Sie beachten, dass die Namen der Tags case-sensitiv sind. Schreibfehler verursachen Fehler!

Tipp:
Erstellen Sie das Projekt zuerst im Selektieren-Register und speichern es anschliessend. Danach können Sie die Projektdatei manuell editieren.

Dies ist der Layout einer Projektdatei, bei welcher alle Tabellen von zwei Dateien miteinander verglichen werden, ohne irgendwelchen Optionen. Sie erkennen dies anhand des <Pairs>-Tag, weil dort keine individuellen Eigenschaften zugewiesen wurden.

```
<?xml version="1.0" standalone="yes"?>
<Synkronizer>
   <Project ID="1">
      <Settings>
         <CompareType>0</CompareType>
         <Formats>0</Formats>
         <Filters>0</Filters>
         <FilterTolerance>0</FilterTolerance>
         <FilterPattern/>
```

```
            <ReportType>0</ReportType>
            <HighlightType>1</HighlightType>
            <ShowHide>0</ShowHide>
        </Settings>
        <Files>
            <Filename0>C:\Folder\SubFolder\File 1.xls</Filename0>
            <Filename1>C:\Folder\SubFolder\File 2.xls</Filename1>
        </Files>
        <Pairs>
            <MatchType>1</MatchType>
            <MatchInclude>2</MatchInclude>
        </Pairs>
    </Project>
</Synkronizer>
```

Für ein Projekt mit individuellen Optionen sieht der Layout wie folgt aus:
(MatchType wird auf 0 gesetzt, da die <Pair>-Objekte individuelle Eigenschaften wie RangeAddr oder DBKeys enthalten)

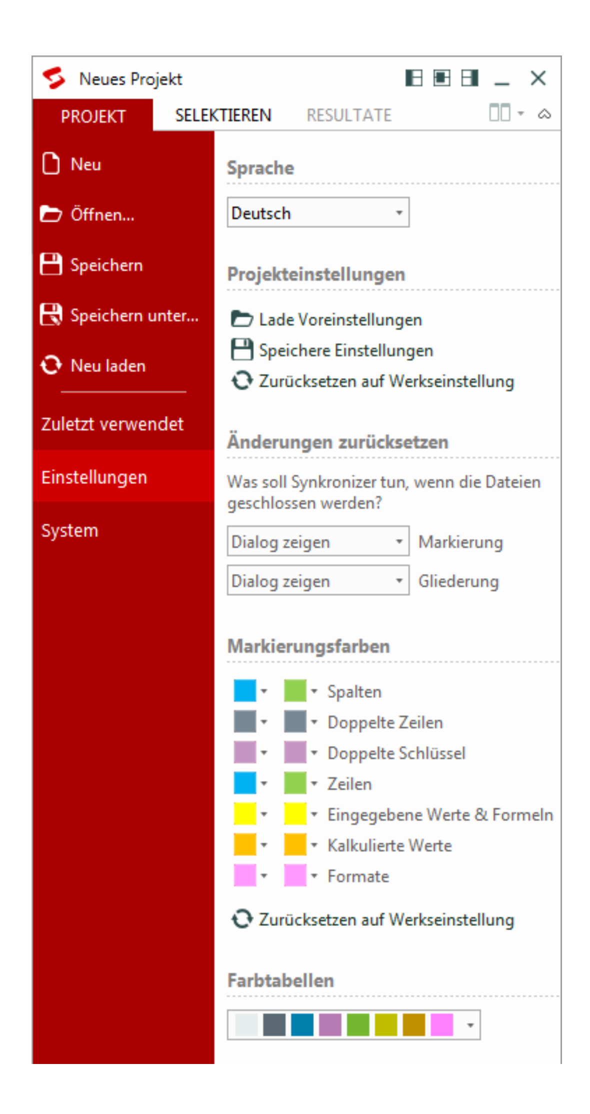
Die <Pair>-Tags müssen sequentiell mit dem ID Attribut nummeriert werden.

```
<?xml version="1.0" standalone="yes"?>
<Synkronizer>
    <Project ID="1">
        <Settings>
            <CompareType>0</CompareType>
            <Formats>0</Formats>
            <Filters>0</Filters>
            <FilterTolerance>0</FilterTolerance>
            <FilterPattern/>
            <ReportType>0</ReportType>
            <HighlightType>1</HighlightType>
            <ShowHide>0</ShowHide>
        </Settings>
        <Files>
            <Filename0>C:\Folder\SubFolder\File 1.xls</Filename0>
            <Filename1>C:\Folder\SubFolder\File 2.xls</Filename1>
        </Files>
        <Pairs>
            <MatchInclude>3</MatchInclude>
            <MatchType>0</MatchType>
            <PairCount>3</PairCount>
            <Pair ID="1">
                <SheetName0>Addresses</SheetName0>
                <SheetName1>Addresses</SheetName1>
            </Pair>
            <Pair ID="2">
                <SheetName0>Controlling</SheetName0>
                <SheetName1>Controlling</SheetName1>
                <RangeAddr0>$A$4:$AC$75</RangeAddr0>
                <RangeAddr1>$A$4:$AC$75</RangeAddr1>
                <DBRow>5</DBRow>
```

```
            <DBKeys>1</DBKeys>
            <DBOptionsMask>14</DBOptionsMask>
         </Pair>
         <Pair ID="3">
            <SheetName0>Budget</SheetName0>
            <SheetName1>Budget</SheetName1>
         </Pair>
      </Pairs>
   </Project>
</Synkronizer>
```

4.2 Einstellungen

In diesem Menü können allgemeine Einstellungen vorgenommen werden.



Folgende Einstellungen können getätigt werden:

Sprache

Mit diesem Feld kann die Sprache bestimmt werden. Sollte eine Sprache nicht vorhanden sein, offerieren wir Ihnen eine Developer Version für eine Übersetzung des Synkronizer GUIs (Grafische Benutzeroberfläche). Detaillierte Instruktionen werden auf Anfrage gemacht.

Projekteinstellungen

Hier können Sie Ihre persönlichen Projekteinstellungen speichern, neu laden oder zurücksetzen. Folgende Optionen stehen zur Auswahl:

- **Lade Voreinstellungen**
  Ihre bevorzugte, persönliche Voreinstellung wird geladen (siehe auch "Aktuelle Einstellungen als Voreinstellung speichern").

- **Speichere Einstellungen**
  Mit dieser Schaltfläche können Sie Ihre persönlichen Wunscheinstellungen als Voreinstellung speichern. Sämtliche Einstellungen der Kategorien Inhalte, Formate, Filter, Bericht, Markieren und Gliederung werden gespeichert.

- **Zurücksetzen auf Werkseinstellung**
  Sämtliche Einstellungen des Selektieren-Registers werden auf die Werkseinstellung zurückgesetzt.

Änderungen zurücksetzen

Eine Stärke des Synkronizers ist, dass er die Differenzen in den Tabellen sauber markieren und unerwünschte Daten mittels der Gliederung ausblenden kann. Das Problem ist aber, dass diese Änderungen unter Umständen in den Exceldateien bleiben und nicht zurückgesetzt werden.

In diesem Bereich können Sie bestimmen, was mit den Markierungs- und Gliederungs-Änderungen passieren soll. Folgende Optionen stehen zur Verfügung:

- **Dialog zeigen**
  Beim Schliessen der Exceldatei wird ein Dialogfeld gezeigt, in welchem Sie entscheiden können, ob die Änderungen zurückgesetzt werden sollen oder nicht.

- **Nicht zurücksetzen**
  Die Synkronizer-Änderungen werden nicht zurückgesetzt.

- **Immer zurücksetzen**
  Die Synkronizer-Änderungen werden zurückgesetzt. Ihre Exceldateien befinden sich somit im selben Status wie vor dem Vergleich (ausser den transferierten Differenzen).

Farbtabellen / Markierungsfarben

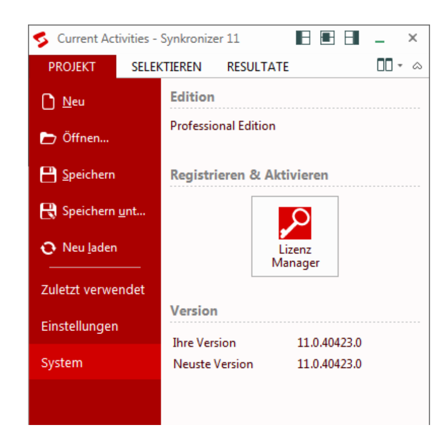
Falls Ihnen die Farben, mit denen der Synkronizer die Differenzen markiert, nicht gefallen, haben Sie hier die Möglichkeit, Ihre eigenen Farben festzulegen. Mit der Schaltfläche "Zurücksetzen auf Werkseinstellung" werden die Farben wieder auf die ursprünglichen "Synkronizer"-Farben

rückgesetzt.

Keine Farbe verwenden
Wenn gewisse Differenztypen nicht markiert werden sollen, dann wählen Sie einfach den gewünschten Differenztyp und selektieren "Keine Farbe verwenden".

4.3 System

In diesem Menü sehen Sie die Informationen über Ihre Synkronizer-Lizenz.



Edition
Hier sehen Sie, welche Edition installiert ist.

Registrieren und Aktivieren
Im Lizenz Manager können Sie genauere Informationen über Ihre Lizenz entnehmen und die Software registrieren, aktivieren und deaktivieren. Ebenso können bei einer Netzwerklizenz die Netzwerkbenutzer verwaltet werden.

Version
Hier können Sie prüfen, ob die neuste Version (oder Build) installiert ist. Falls nicht, erscheint eine Schaltfläche zum Download der aktuellsten Versioon.

Lizenz Manager
Mit der Schaltfläche "Lizenz Manager" wird der Lizenz Manager gestartet. Im Lizenz Manager sind weitere Informationen über die installierte Lizenz ersichtlich.

4.3.1 Lizenz Manager

Im Lizenz Manager können sämtliche Einstellungen über die Synkronizer-Lizenz vorgenommen werden.



Die folgenden Informationen/Aktionen sind ersichtlich/können festgelegt werden:

- detaillierte Informationen über Lizenz
- Lizenz registrieren/aktivieren
- Lizenz deaktivieren
- Netzwerkbenutzer verwalten (betrifft nur Netzwerklizenz)

Part V

5 Installation, De-Installation

Synkronizer 11 ist ein COM-Addin für Excel 2013, 2010 und 2007. Synkronizer funktioniert sowohl im 32, als auch 64-Bit-Modus.

Installation

1. Downloaden Sie die neueste Version des Synkronizers.

2. Melden Sie sich unter Windows **mit Administratorenrechten** an.

3. Schliessen Sie Excel.

4. Doppelklicken Sie auf die Installations-Datei SetupSynkronizer.11.0.xxx.exe.

5. Starten Sie Excel.

6. Das Synkronizer Startsymbol befindet sich im Add-ins Menü



7. Registrieren Sie die Software

Administratorenrechte

Wenn Sie keine Administratorenrechte besitzen, können Sie die Software nicht installieren.

Installationsmethoden

Es gibt folgende Installationsmethoden:

- **Standardinstallation**
  Diese Installationsmethode wird für **Einzelplatzlizenzen** und **Mehrplatzlizenzen** verwendet.

- **Netzwerkinstallation**
  Der Administrator muss zuerst ein Netzwerkverzeichnis freigeben und dort die Lizenzdatei ablegen, danach kann er Synkronizer normal installieren. Im Lizenzmanager des Synkronizers muss er dann noch das Netzwerkverzeichnis festlegen.

Verzeichnisse

Alle Dateien werden in der Regel in diesem Verzeichnis installiert:
C:\Program Files[(x86)]\Synkronizer\Synkronizer 11\

Die "Cached Assemblies" befinden sich in einem Unterverzeichnis in
C:\Users\UserName\AppData\Local\Assembly\dl3

Die Konfigurationsdateien befinden sich in einem "Hash"-Unterverzeichnis in
C:\Users\Username\AppData\Local\Microsoft_Corporation\

Die Lizenzdatei befindet sich hier:
C:\ProgramData\Synkronizer\Synkronizer 11

Das Netzwerkverzeichnis wird im Registry wie folgt gespeichert:
Schlüssel (32-Bit Windows): HKLM\Software\Synkronizer\Synkronizer 11
Schlüssel (64-Bit Windows): HKLM\Software\Wow6432Node\Synkronizer\Synkronizer 11
Name: LicensePath (String)
Wert: e.g. \\server\path\license.xml

Softwareanforderungen - Prerequisites

- Microsoft .NET Framework 4.0
- Visual Studio Tools for Office Runtime 2010

Das Setup-Programm sucht nach diesen Programmen und prüft, ob die aktuellste Version vorhanden ist. Je nach Status, werden die Software installiert/aktualisiert.

5.1 Deinstallation

Wenn Sie Synkronizer entfernen möchten, gehen Sie wie folgt vor:

1. Öffnen Sie in Windows die Systemsteuerung.
2. Wählen Sie „Programme und Funktionen"
3. Selektieren Sie „Synkronizer 11" und klicken auf „deinstallieren".

5.2 Netzwerkinstallation

Bei einer Netzwerkinstallation muss die **Lizenzdatei** in einem **freigegebenen Netzwerkerzeichnis** (**Shared Folder**) gespeichert werden. Sämtliche Benutzer, die mit Synkronizer arbeiten, müssen Leserechte auf dieser Datei haben. Alle anderen Dateien dürfen sich in einem beliebigen Verzeichnis befinden.

Die Netzwerklizenz

Die Netzwerklizenz ist eine einzelne Datei, die sämtliche Software-, Hardware- und Benutzerdaten enthält. Die Netzwerklizenz wird einmalig installiert und ist fix an das Netzwerkverzeichnis gebunden. Danach können die einzelnen Benutzer PCs/Arbeitsplätze (welche mit Synkronizer arbeiten) hinzugefügt oder entfernt werden, ohne dass die Lizenz nochmals aktiviert werden muss.

Erst-Installation

1. Erstellen Sie einen **freigegebenen** Ordner in einem Netzwerk (**Shared Folder**)

2. Erstellen Sie in diesem Ordner eine Textdatei mit dem Namen **License.xml**.

3. Schreiben Sie folgenden Text in die Lizenzdatei:
```
<?xml version="1.0" encoding="UTF-16" standalone="yes"?>
<SynkLic />
```

4. Speichern Sie die Datei.

5. Starten Sie Excel.

6. Starten Sie Synkronizer und öffnen den Lizenzmanager mit dem folgenden Befehl:
   Projekt » System » Lizenz Manager

   es erscheint folgendes Formular



7. Klicken Sie auf das Verzeichnis-Symbol (siehe Pfeil) und wählen das Netzwerkverzeichnis.

   Anmerkung:
   Das Netzwerkverzeichnis muss im UNC (Uniform Naming Convention) angegeben werden und sieht wie folgt aus:

   \\Servername\Freigabename\Pfad\License.xml

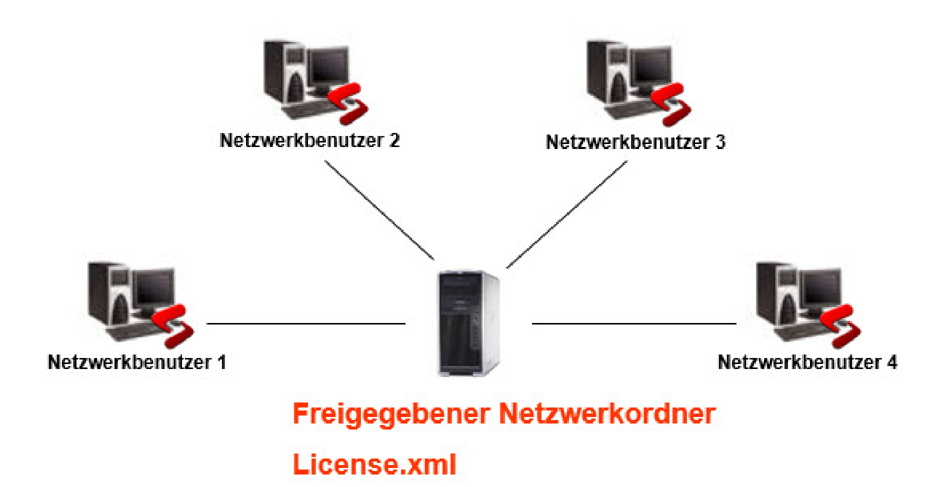
8. Registrieren und aktivieren Sie die Software

9. Fügen Sie die Netzwerk-Benutzer hinzu, die mit Synkronizer arbeiten (siehe Kapitel weiter unten)

Installationen der Netzwerkbenutzer

Die Installation der Netzwerkbenutzer ist denkbar einfach. Die einzelnen Netzwerkbenutzer installieren Synkronizer normal auf ihrem lokalen Computer und verweisen im Lizenzmanager auf die Lizenzdatei, die sich im Netzwerkordner befindet. Sofern die Software aktiviert und der Netzwerkbenutzer zugewiesen ist, kann der Benutzer sofort mit Synkronizer arbeiten.

Schema Netzwerk



Die Lizenzdatei (License.xlm, rot) befindet sich in einem freigegebenen Netzwerkordner (Shared Folder). Die einzelnen Netzwerkbenutzer installieren Synkronizer normal auf ihrem lokalen Computer und verweisen im Lizenzmanager auf die Lizenzdatei, die sich im Netzwerkordner befindet.

Rechte

Mindestens ein Benutzer muss Schreibrecht auf der Lizenzdatei (License.xml) besitzen. Benutzer mit Schreibrechten können die Software registrieren/aktivieren und die einzelnen (autorisierten) Benutzer verwalten. Benutzer, die nur Leserechte haben, können nur mit der Software arbeiten.

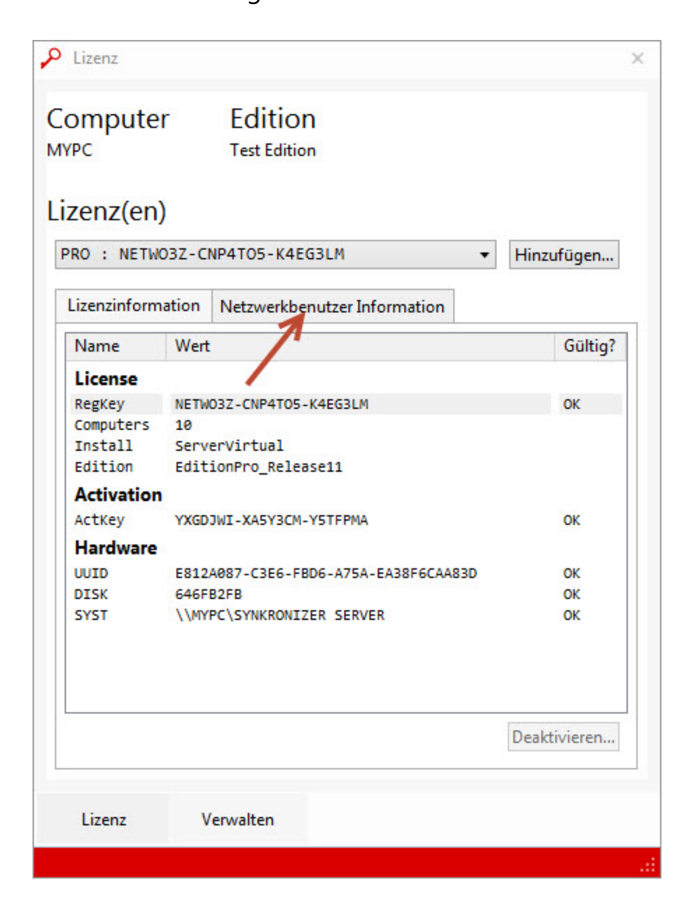
5.2.1 Arbeitsplätze hinzufügen

Sobald die Lizenz aktiviert ist, können Sie die Arbeitsplätze/Anwender hinzufügen, die mit Synkronizer arbeiten sollen. Dies kann von jedem Arbeitsplatz aus geschehen, auf dem Synkronizer installiert ist, solange der Anwender **Schreibrechte** auf der Lizenzdatei besitzt.

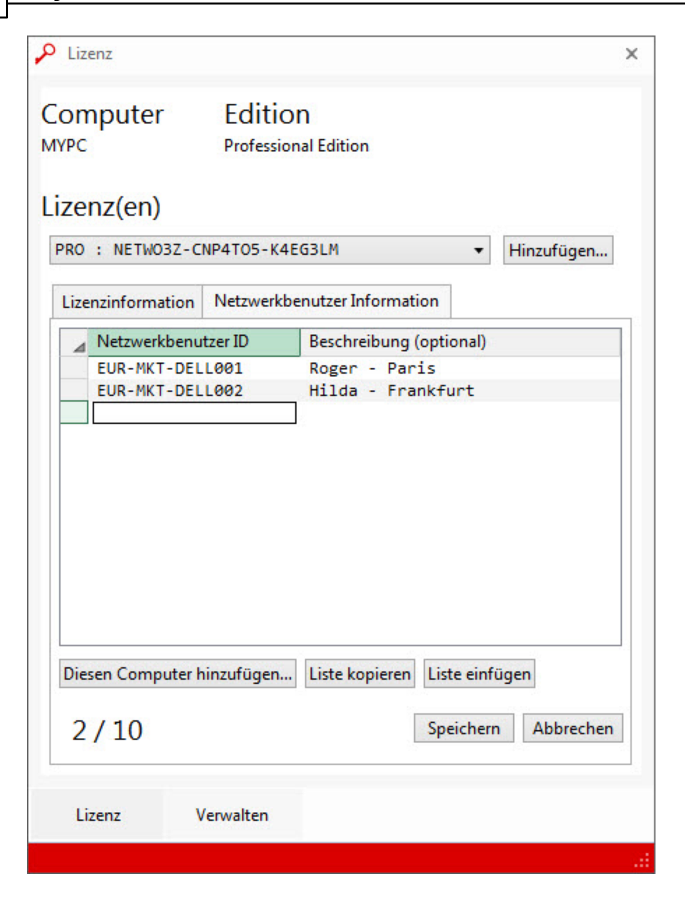
1. Wählen Sie den Projekt » System » Lizenzmanager.

2. Klicken Sie auf "Verwalten"

3. Wählen Sie das Register "Netzwerkbenutzer Information"



4. Nun können Sie die Computer/Anwender hinzufügen, die mit Synkronizer arbeiten sollen. Geben Sie einfach die korrekte Netzwerkbenutzer ID und die Beschreibung in die Eingabefelder ein. Wenn Sie den PC, mit dem Sie arbeiten, hinzufügen möchten, klicken Sie auf "Diesen Computer hinzufügen".



Anmerkung:
Nur der Computername muss eingegeben werden. Das Beschreibungsfeld ist optional zur freien Verfügung.

5. Bestätigen Sie die Mutationen mit der Speichern-Taste.

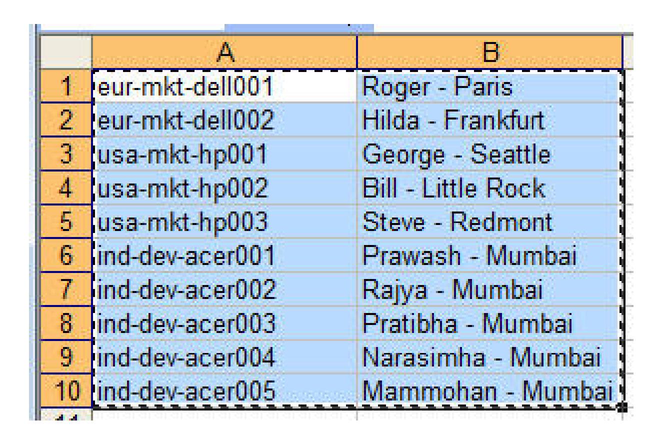
Mehrere Arbeitsplätze hinzufügen

Sie können die Arbeitsplätze/Anwender auch in eine Excel-Liste eingeben und danach in die Lizenzmaske einfügen. Gehen Sie wie folgt vor:

1. Starten Sie Excel

2. Erstellen Sie eine zweispaltige Liste bestehend aus Arbeitsplätzen und Beschreibung

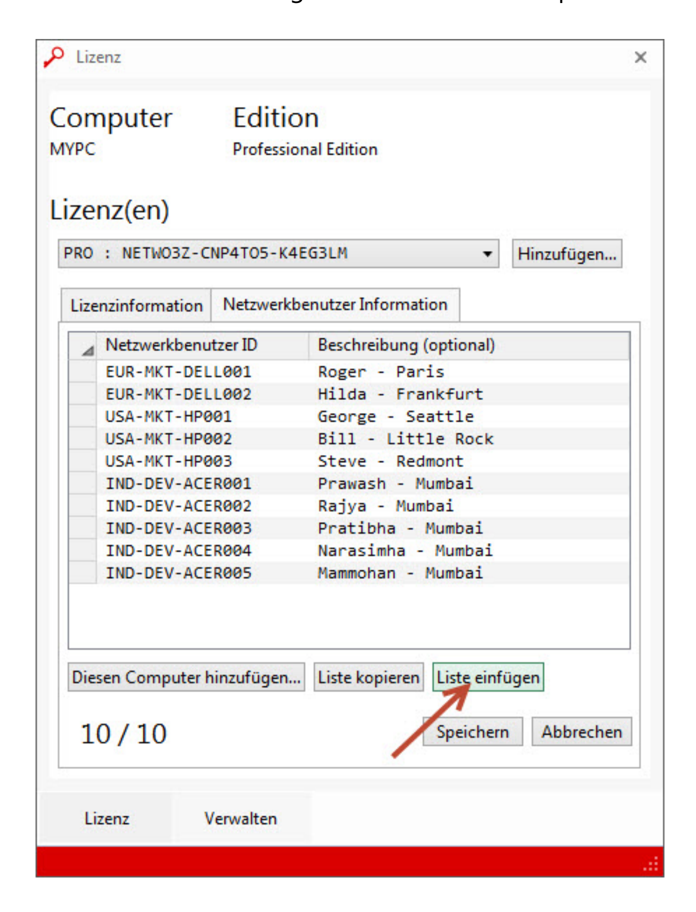
3. Markieren und kopieren Sie die Liste



	A	B
1	eur-mkt-dell001	Roger - Paris
2	eur-mkt-dell002	Hilda - Frankfurt
3	usa-mkt-hp001	George - Seattle
4	usa-mkt-hp002	Bill - Little Rock
5	usa-mkt-hp003	Steve - Redmont
6	ind-dev-acer001	Prawash - Mumbai
7	ind-dev-acer002	Rajya - Mumbai
8	ind-dev-acer003	Pratibha - Mumbai
9	ind-dev-acer004	Narasimha - Mumbai
10	ind-dev-acer005	Mammohan - Mumbai

4. Öffnen Sie die "Verwalten"-Maske des Lizenzformulares.

5. Klicken Sie "Liste einfügen". Sämtliche Arbeitsplätze werden in die Liste eingefügt



5.3 Installation mithilfe eines Skripts

Das Setup-Programm ist ein sogenannter Bootstrapper für die darin enthaltenen Windows Installer Dateien, es kümmert sich auch um die notwendigen Voraussetzungen (.NET Framework und VSTO Runtime)

Kommandozeilenparameter:

/? or /help
zeigt Parameter

/extract:<directory>
extrahiert alle Dateien in <directory>

/listlangs
listet verfügbare Sprachen

/exenoui
Installation wird ohne UI (Benutzerformulare) im Hintergrund ausgeführt

/exebasicui
Installation wird mit Basis UI ausgeführt

/exelang <langId>
Installation wird in gewünschter Sprache ausgeführt

/username
Benutzername des Proxy-Servers

/password
Kennwort des Proxy-Servers

/exelog<path_to_log_file>
Erstellt eine Logdatei im gewünschten Verzeichnis

/exenoupdates
kontrolliert nicht nach einer neueren Version

/passive licensepath=<Server Path + License.xml>
Netzwerkinstallation; definiert den Pfad und Name der Lizenzdatei

<msiOptions>
Optionen des msiexec.exe Paketes

Beispiel:

Installation ohne UI:

Synkronizer.11.0.xxxx.exe /exenoui

Part VI

6 Registrierung

Wenn Sie die Software kaufen, erhalten Sie einen Registrierungsschlüssel. Dieser Schlüssel wird für die Registrierung und Aktivierung der Software benötigt. Der Aktivierungsprozess läuft automatisch ab, es wird nur eine Internetverbindung benötigt. Alternativ kann die Software auch via unserer Internetseite aktiviert werden..

Was passiert während der Aktivierung?

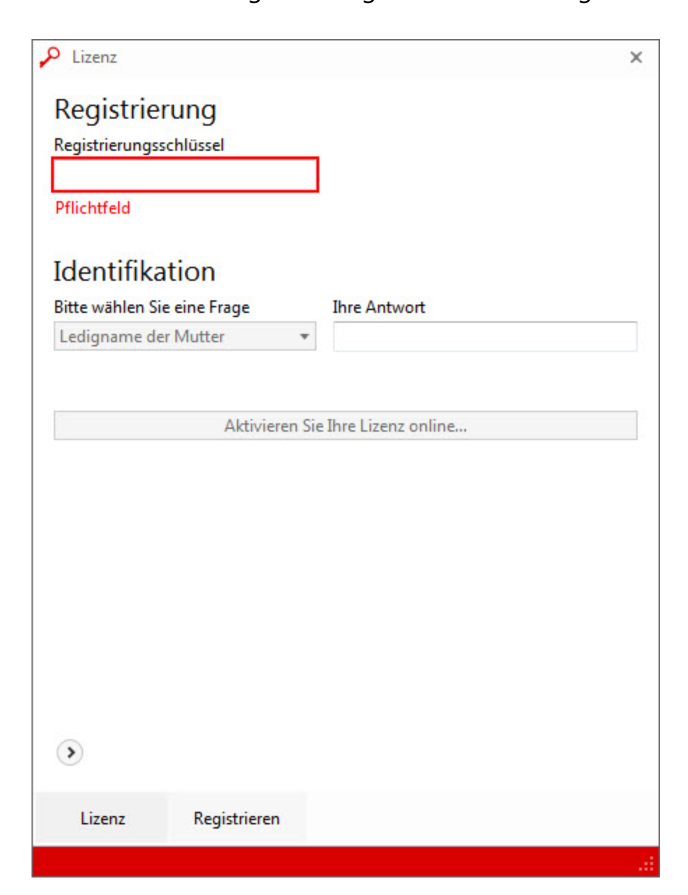
Der Registrierungsschlüssel und ein paar Hardwareinformationen (keine persönlichen Daten!) werden an unseren Aktivierungsserver gesendet. Der Aktivierungsserver erzeugt dann anhand dieser Hardwaredaten einen Aktivierungsschlüssel und retourniert diesen an Ihren Computer. Dieser Aktivierungsschlüssel muss mit den Hardwaredaten übereinstimmen und wird in einer Lizenzdatei gespeichert. Die Lizenzdatei ist somit an den PC verbunden und die Software wird aktiviert.

Die Software wird wie folgt aktiviert:

1. Starten Sie Synkronizer.

2. Öffnen Sie den Lizenzmanager mit dem folgenden Befehl:
   Projekt » System » Lizenz Manager

3. Wählen Sie den Register "Registrieren". Das folgende Formular erscheint:



4. Geben Sie den Registrierungsschlüssel ein. Bei einer Mehrplatzlizenz muss Sie noch zusätzlich die Laufnummer der Lizenz eingegeben werden (rechts nebem dem Registrierungsschlüssel-Feld).

5. Geben Sie die Antwort der Sicherheitsfrage ein.

   Anmerkung:
   Die Sicherheitsfrage ist ein zusätzlicher Schutz, damit niemand ausser Ihnen Ihre Lizenz verwenden kann. Wenn Sie Ihre Lizenz auf einem zweiten PC installieren möchten, müssen Sie ebenfalls die selbe Antwort eingeben. Die Länge der Sicherheitsantwort muss mindestens 5 Zeichen betragen. Folgende Zeichen sind erlaubt: A-Z und 0-9.

6. Klicken Sie "Aktivieren Sie Ihre Lizenz online...". Die Software wird online aktiviert.

   Falls eine Internetverbindung vorliegt, werden die Daten direkt an den Aktivierungsserver übermittelt und die Software wird aktiviert. Liegt hingegen keine Internetverbindung vor, müssen Sie die Software manuell aktivieren. Genaue Instruktionen entnehmen Sie diesem Kapitel.

Falls Sie Probleme mit der Registrierung/Aktivierung haben sollten, dann lesen Sie das Kapitel Probleme mit Registrierung.

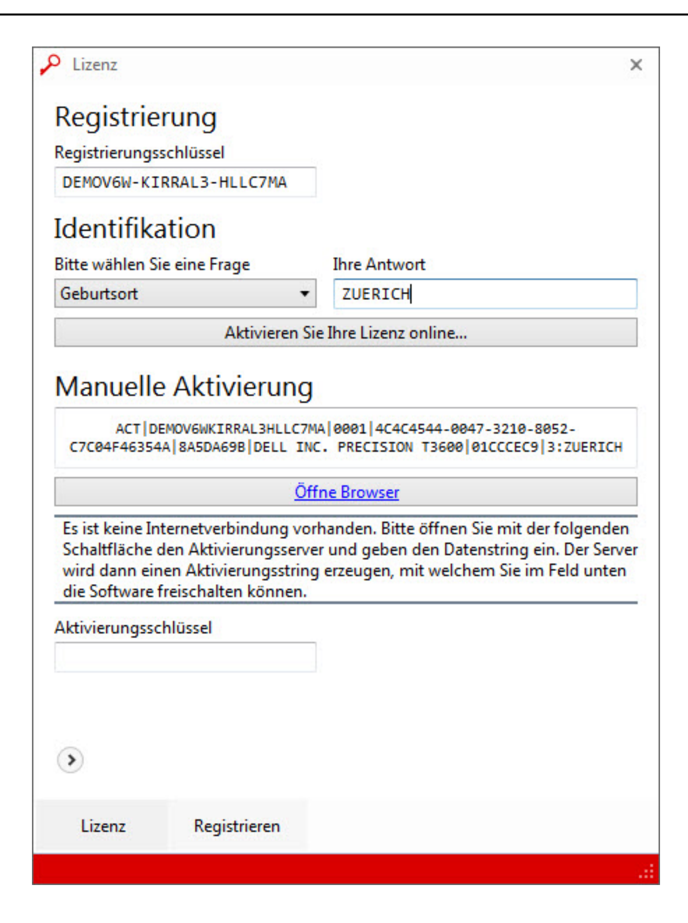
6.1 Manuelle Aktivierung

Eine manuelle Aktivierung ist nur notwendig, wenn Ihr Computer keine Internetverbindung aufweist. In diesem Fall müssen Sie die Aktivierungsdaten manuell auf unserer Internetseite eintragen, damit Sie den Aktivierungsschlüssel erhalten.

Dieses Kapitel erklärt, wie die Software manuell aktiviert wird.

Die Registrierungsanweisungen 1-6 sind im Kapitel Registration beschrieben.

7. Wenn keine Internetverbindung hergestellt werden konnte, wird folgende Maske angezeigt:



Es wird ein Datenstring erzeugt, welcher die Hardwaredaten Ihres Computers enthält.

8. Sie müssen nun diesen Datenstring kopieren, damit Sie ihn später auf der Internetseite wieder einfügen können. Öffnen Sie hierfür z.B. den Texteditor und fügen den Datenstring mit der Tastaturkombination "CTRL-V" ein.

9. Klicken Sie auf den Link "Öffne Browser" oder rufen folgende Internetseite auf: http://www.synkronizer.de/aktivierungsschluessel
   Es erscheint die Aktivierungsseite:



10. Geben Sie nun im Feld 1) den Datenstring ein und klicken auf das Aktivierungssymbol, damit der Aktivierungsschlüssel erzeugt wird.

2) Aktivierungs-/Deaktivierungsschlüssel generieren

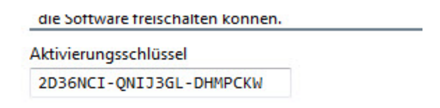


3) Aktivierungsschlüssel in Lizenzformular eingeben

Bitte kopieren Sie den Aktivierungsschlüssel und fügen ihn im Eingabefeld des Lizenzformulares ein.

Aktivierungsschlüssel: 2D36NCI-ZHKYYJD-B4AYICK

11. Notieren Sie sich den Aktivierungsschlüssel und rufen das Lizenzformular auf.



12. Geben Sie den Aktivierungsschlüssel ein.

13. Die Software ist aktiviert.

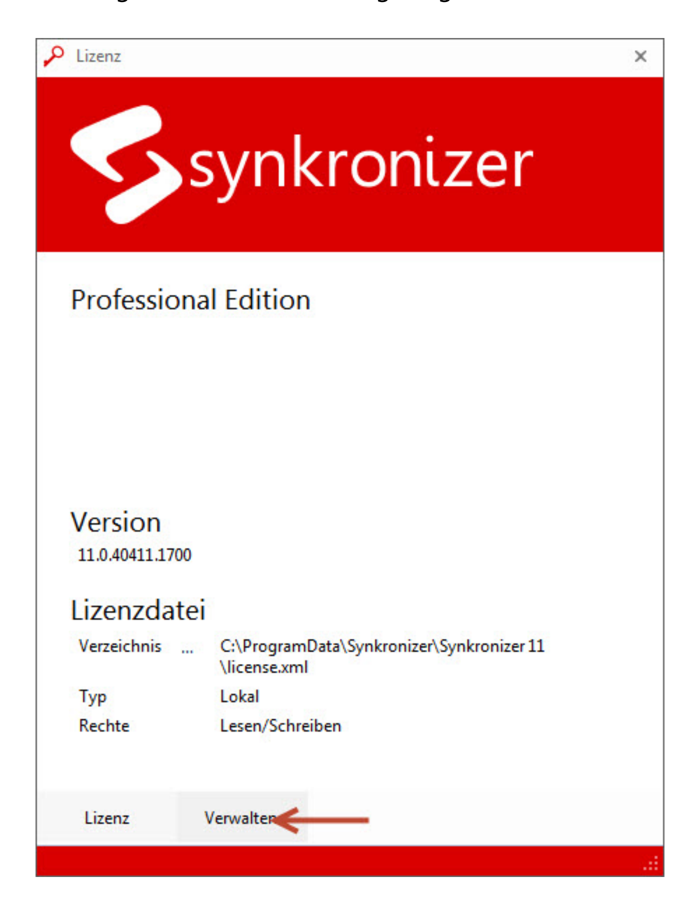
Falls Sie Probleme mit der Registrierung/Aktivierung haben sollten, dann lesen Sie das Kapitel Probleme mit Registrierung.

6.2 Deaktivieren

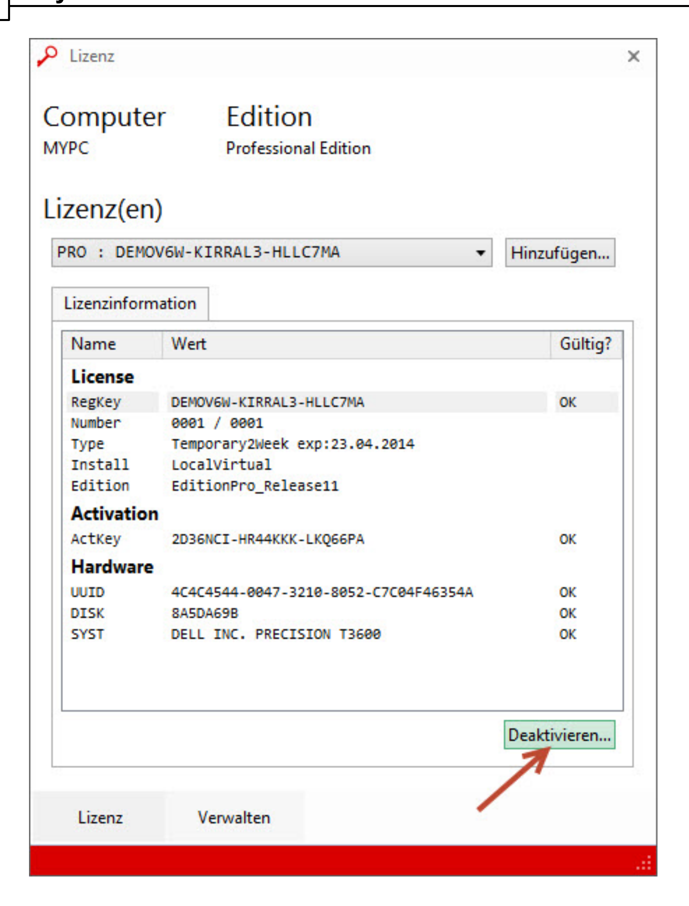
Falls Sie einen anderen Computer gekauft haben oder ihr alter Computer defekt sein sollte, können die Software auf einen anderen Computer transferieren. Sie müssen einfach die Lizenz zuerst deaktivieren, bevor Sie sie auf dem neuen Computer installieren können.
Gehen Sie wie folgt vor:

1. Starten Sie Synkronizer.

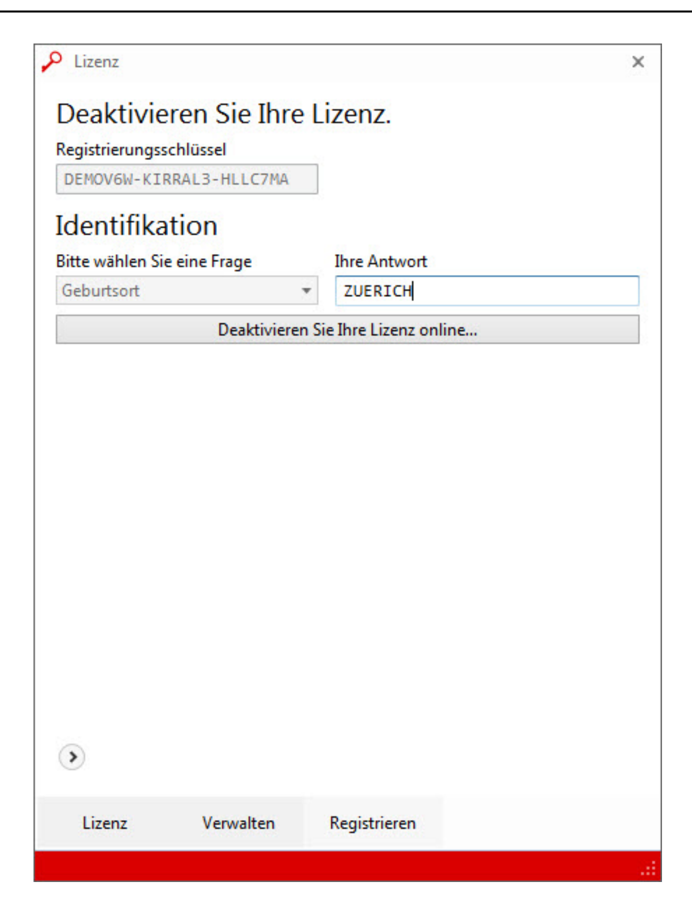
2. Wählen Sie den Befehl "Projekt » System » Lizenzmanager".
   Das folgende Fenster wird angezeigt:



3. Wählen Sie den Register "Verwalten".



4. Klicken Sie auf "Deaktivieren". Es erscheint das "Registrieren"-Formular



5. Geben Sie die Sicherheits-Antwort ein und klicken auf die Deaktivieren-Schaltfläche. Die Software ist deaktiviert.

Sollten Sie über keinen Internetanschluss verfügen, dann müssen Sie die Software manuell deaktivieren. Eine manuelle Deaktivierung funktioniert genau gleich wie eine manuelle Aktivierung, nur mit dem Unterschied, dass die Software deaktiviert wird. Lesen Sie dieses Kapitel.

Part VII

7 Probleme / Fehler

Synkronizer 11 wird nicht in der Symbolleiste angezeigt

Wenn Excel abstürzt, kann es sein, dass das Synkronizer Symbol nicht mehr in der Symbolleiste angezeigt wird. Gehen Sie wie folgt vor um das Problem zu lösen:

1. Klicken Sie auf das Microsoft Office-Symbol » Excel-Optionen » Add-Ins

2. Kontrollieren Sie, ob Synkronizer 11 unter "Deaktivierte Anwendungs-Add-Ins" gelistet (sollte korrekt unter "Aktive Anwendungs-Add-Ins" erscheinen).

3. Um Synkronizer wieder zu aktivieren, markieren und entfernen Sie Synkronizer vom "deaktiven" Bereich. Synkronizer erscheint dann unter "Aktive Anwendungs-Add-Ins" oder "Inaktive Anwendungs-Add-Ins". Falls Synkronizer unter "Inaktive Anwendungs-Add-Ins" gelistet sein sollte, müssen Sie die ComAdin-Schaltfläche wählen und dann Synkronizer aktivieren.

Jetzt sollte das Synkronizer Logo wieder in der Symbolleiste sichtbar sein. Falls nicht, müssen Sie folgende Schritte ausüben:

1. Öffnen Sie mit dem Windows Explorer das folgende Verzeichnis:
   C:\Program Files\Synkronizer\Synkronizer 11

2. Doppelklicken Sie auf die Datei **AddinLoad.bat**.

3. Wählen Sie die Option **1) Load in Current User (HKCU)**

4. Schliessen Sie das Fenster.

Excel 2013 - Geschützte Dateien

Excel 2013 ist sehr langsam, wenn Sie mit kennwortgeschützten Dateien arbeiten.

Excel 2007 - Digitale Signatur ist ungültig

Excel 2007 akzeptiert die neuen digitalen Signaturen nicht. Die neuen Signaturen werden erst ab Office 2010 unterstützt. (http://support2.microsoft.com/lifecycle/?p1=8753)

Gehen Sie wie folgt vor um das Problem zu beheben:

- Stellen Sie sicher, dass das neuste Service Pack von Excel 2007 installiert ist:
  Excel 2007 SP3 – (12.0.6683.5002) mit MSO(12.0.6683.5000)

- Stellen Sie sicher, dass "XL Consulting GmbH” unter "Vertrauenswürdige Herausgeber" gelistet ist.
  Sie können das wie folgt kontrollieren:
  - Klicken Sie auf die Windows Starttaste
  - Geben Sie "cmd" ein
  - Geben Sie "certmgr" ein - das Vertrauenscenter wird geöffnet
  - Wählen Sie "Vertrauenswürdige Herausgeber"

Wenn "XL Consulting GmbH" nicht in der Liste ist, müssen Sie wie folgt vorgehen:
- schliessen Sie Excel
- Öffnen Sie den Windows Explorer
- Wählen Sie das Synkronizer Installationsverzeichnis ("C:\Programme\Synkronizer\Synkronizer 11")
- Klicken Sie mit der rechten Maustaste auf synkronizer.dll
- Wählen Sie Eigenschaften
- Wählen Sie "Digitale Signaturen"
- Wählen Sie "XL Consulting GmbH”
- Wählen Sie die "Details” Schaltfläche
- Wählen Sie "Zertifikat anzeigen”
- Wählen Sie "Zertifikat installieren”

--- "Zertifikatimport Assistent” Dialog ---
- Wählen Sie "Weiter"
- Wählen Sie "Alle Zertifikate in folgendem Speicher speichern
- Wählen Sie "Vertrauenswürdige Herausgeber"
- Wählen Sie "Ok"
- Wählen Sie "Weiter"
- Wählen Sie "Fertig stellen"

- Schliessen Sie alle offenen Fenster
- Starten Sie Excel

7.1 Support

Sollten Sie Probleme mit Synkronizer haben (Fehlermeldungen, Installationsprobleme, usw.), dann gehen Sie wie folgt vor:

1. Erstellen Sie eine neue Email.

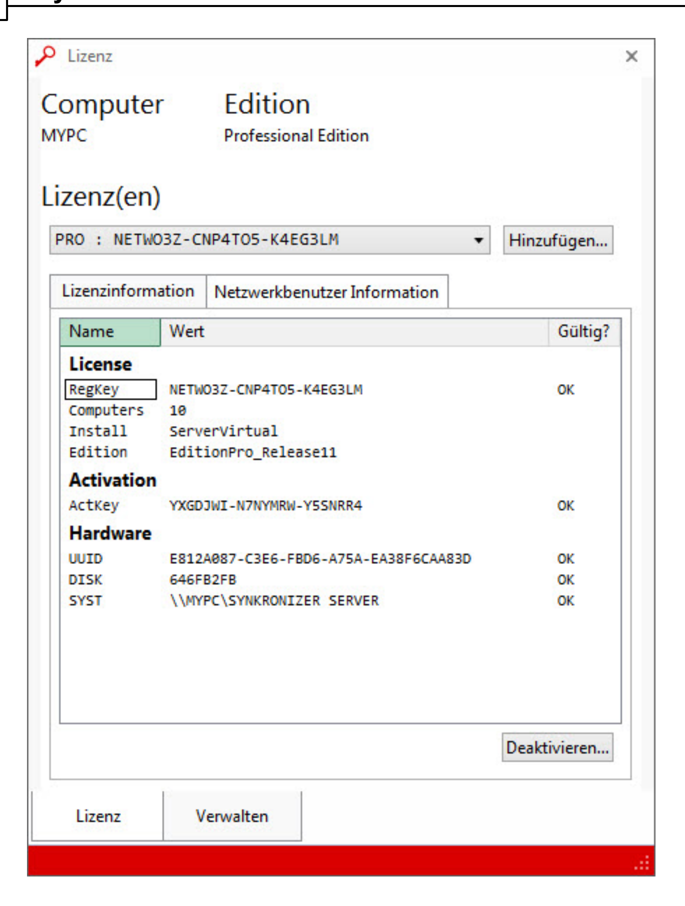
2. Starten Sie Synkronizer.

3. Wählen Sie Projekt » System » Lizenzmanager
   Der Lizenzmanager wird geöffnet.

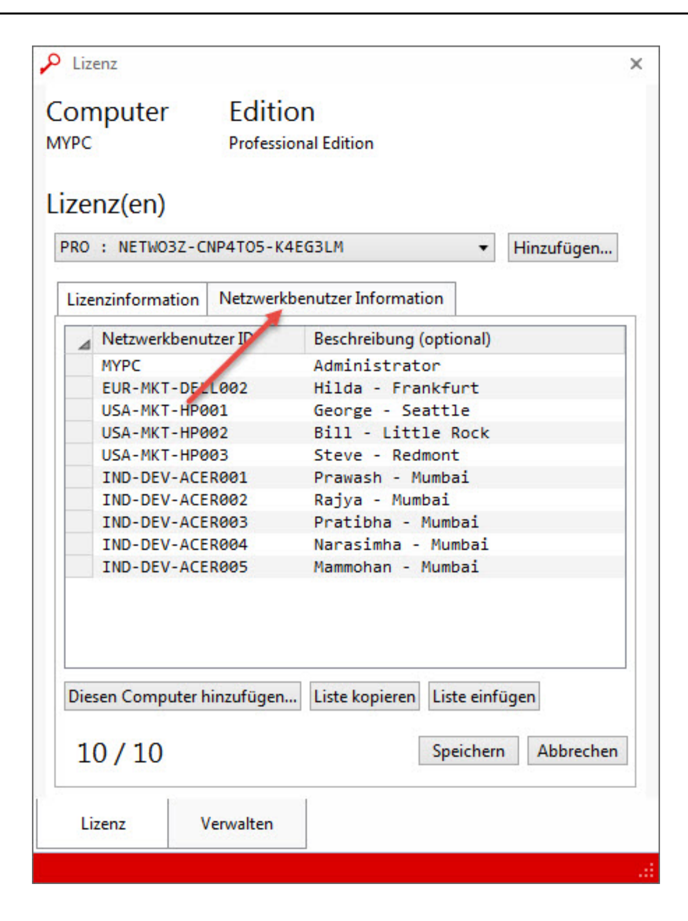
4. Erstellen Sie einen Screenshot des Lizenzmanagers:



5. Klicken Sie auf das Register "Verwalten". Erstellen Sie einen Screenshot dieses Bildschirms:



6. Falls Sie mit einer Netzwerklizenz arbeiten, klicken Sie auf "Netzwerkbenutzer Information" und erstellen einen Screenshot:



7. Fügen die Datei **license.xml** der E-Mail hinzu. Die Lizenzdatei befindet sich in dem folgenden Verzeichnis;

   Lokale Lizenz
   C:\ProgramData\Synkronizer\Synkronizer 11

   Netzwerklizenz
   Freigegebenes Netzwerkerzeichnis (Shared Folder)

8. Kontrollieren Sie, ob eine Fehlerdatei im temporären Windowsverzeichnis erstellt wurde. Starten Sie den Windows Explorer und geben das Verzeichnis **%TEMP%** ein. Falls sich in diesem Verzeichnis eine Datei mit dem Namen **synkronizerError.txt** befindet, fügen Sie diese ebenfalls der E-Mail hinzu.

9. Senden Sie die dann die Screenshots und Dateien an support@synkronizer.com. Erwähnen Sie zusätzlich, was für für Beobachtungen Sie im Zusammenhang mit Synkronizer gemacht haben (z.B. eigenartiges Verhalten, ist der Fehler replizierbar?, uzsw.). Das hilft uns bei der Lösung Ihres Falls. Unser Supportteam wird Ihnen innerhalb 24h eine Antwort senden.

Part VIII

8 Developer Edition

8.1 Visual Basic for Applications (VBA)

With the **Developer edition** of Synkronizer, you are able to create **recurring, standardized** and/or **complex comparisons**. To access the automation functionality of Synkronizer, you need to have knowledge about **Visual Basic for Applications (VBA)**. With VBA, you are able to you create your own powerful comparison macros.

We have provided some simple **VBA Examples** and a **"Wrapper" Procedure** to simplify you work.

To test the examples, download the example file **synk11vba.zip** from our server, unzip it and copy the files to a folder of preference.

You can also manually test the examples by using the following steps:

1. Ensure that **Synkronizer 11 Developer** is installed.
2. Start Excel
3. Open a (new) workbook
4. Open the VB Editor (keyboard shortcut Alt-F11)
5. Select your workbook in the Project Explorer window
6. Select menu Tools/References, scroll to **'Synkronizer 11.0 Object Library'**, check it, and press ok.
7. Select menu Insert/Module
8. Copy the **Helper Functions** into this module.
9. Select menu Insert/Module
10. Copy the **VBA Examples** into this module.

The **Object Model Reference** of Synkronizer 11 is described in the following chapters.

8.1.1 Object Model Reference

Object Model Reference of Synkronizer 11

The object model of Synkronizer 11 looks as follows:

Excel
- **ComAddins("Synkronizer.AddIn")**
  - **Object = Synkronizer.Application**
    - **Project**
      - **Files**
      - **Pairs**
        - **Pair**
          - **Results**
      - **Settings**

8.1.1.1 Application Object

Top level object. Entry point for all automation.

Methods

Name	Description
NewProject	Returns a project with empty files, default settings and manual matching
OpenProject	Creates a new project with empty files, default settings and manual matching.

Properties

Name	Description
ActiveProject	Project that is currently active.
DisplayUI	Determines if user interface is displayed or not.

Remarks

The application object is only accessible when you have a licensed Developer Edition

Example

To access the Synkronizer Application, you should do the following:.

```
Dim cai As COMAddIn
Dim snk As Synkronizer.Application

Set cai = Application.COMAddIns("Synkronizer.Addin")
If Not cai.Connect Then
  cai.Connect = True
End If
Set snk = cai.Object
```

8.1.1.1.1 Methods

8.1.1.1.1.1 New Project

Creates a new project with empty files, default settings and manual matching.

Syntax

expression.**NewProject**
expression. A variable representing an **Application** object.

Parameters

none

Return Value
none

Remarks
If another project is active, it will be closed first.

8.1.1.1.1.2 OpenProject

Opens a project as defined in specified xml file.

Syntax
expression.**OpenProject*(file name)***
expression. A variable representing an **Application** object.

Parameters

Name	Type	Description
file name	String	Path and file name of the project file. (File extension is '.xml').

Return Value
none

Remarks
If another project is active it will be closed first.

8.1.1.1.2 Properties

8.1.1.1.2.1 ActiveProject

Returns the project that is currently active. **Read only** .

Syntax
expression.**ActiveProject**
expression. A variable representing an **Application** object.

8.1.1.1.2.2 DisplayUI

Shows or hides the Synkronizer user interface.

Syntax
expression.**DisplayUI**
expression. A variable representing a **Application** object.

8.1.1.2 Project Object

The project is the central object for working with Synkronizer.

Methods

Name	Description
Close	Closes the current project. Optionally closes the source, target and (if available) difference report files
Execute	Executes the compare process.
Save	Saves the project's configuration to the specified path.

Properties

Name	Description
Files	Container for the Files (workbooks) of the project.
IsDisposed	Project was deactivated and internal object set to nothing.
IsValid	Checks if files are valid and pairs exist and are all valid.
Pairs	Container for the Pairs (worksheets) of the project.
ReportWorkbook	Report workbook (when created).
Results	Container for the (combined) Results of all Pairs.
Settings	Container for all Settings of the project.

Events

Name	Description
Progress	Raised during comparison execution to indicate running subprocess.

8.1.1.2.1 Methods

8.1.1.2.1.1 Close

Closes the project results. Optionally closes the workbooks (without saving!)

Syntax

expression.**Close(*CloseFiles,DisplayUndo*)**
expression. A variable representing a **Project** object.

Parameters

Name	Type	Description
CloseFiles	Boolean	Closes the compared files (workbooks).
DisplayUndo	Boolean	Optional. If true and highlighting or outline are activated, the user will be given the option to undo

Return Value

none

Remarks

If you want the files to be saved (with possible highlighting and outline), you must do so prior to calling the Close method.

8.1.1.2.1.2 Execute

Executes the compare process.

Syntax

*expression.***Execute**
expression. A variable representing a **Project** object.

8.1.1.2.1.3 Save

Saves the project's configuration to the specified path.

Syntax

*expression.***Save(*FilePath*)**
expression. A variable representing a **Project** object.

Parameters

Name	Type	Description
FilePath	String	Path and file name of the project file. (File extension is '.xml').

Return Value

none

8.1.1.2.2 Properties

8.1.1.2.2.1 Files

Container for the Files (workbooks) of the project. **Read only Files** object.

Syntax

*expression.***Files**
expression. A variable representing a **Project** object.

8.1.1.2.2.2 IsDisposed

This property checks if the project was deactivated and internal object set to nothing. **Read only Boolean** property,

Syntax

*expression.****IsDisposed***
expression. A variable representing a **Project** object.

8.1.1.2.2.3 IsValid

Checks if files are valid and pairs exist and are all valid. **Read only Boolean** property,

Syntax

*expression.**IsValid***

expression. A variable representing a **Project** object.

8.1.1.2.2.4 Pairs

Container for the pairs (worksheets) of the project. **Read only** **Pairs** object.

Syntax

*expression.***Pairs**

expression. A variable representing a **Project** object.

8.1.1.2.2.5 ReportWorkbook

Report workbook (when created). **Read only Workbook** object.

Syntax

*expression.***ReportWorkbook**

expression. A variable representing a **Project** object.

8.1.1.2.2.6 Results

Container for the (combined) Results of all pairs. **Read only** **Results** object.

Syntax

*expression.***Results**

expression. A variable representing a **Project** object.

8.1.1.2.2.7 Settings

Container for all settings of the project. **Read only** **Settings** object.

Syntax

*expression.***Settings**

expression. A variable representing a **Project** object.

8.1.1.2.3 Events

8.1.1.2.3.1 Progress

Raised during compare execution to indicate running subprocess.

Remarks

You can use this event to display a progress indicator.

8.1.1.3 Files Object

Container for the two files (workbooks) to compare.

Methods

Name	Description
Load	Loads the two files from disk.
Save	Saves the source file and the target file.

Properties

Name	Description
file name	Returns the full path and file name of either of the files.
IsValid	Returns true if both files are loaded and can be compared.
Window	Returns the window object of either of the files.
Workbook	Returns the workbook object of either of the files.

Events

Name	Description

none

8.1.1.3.1 Methods

8.1.1.3.1.1 Load

Loads the two files from disk.

Syntax

expression.**Load*(file name0,file name1)***

expression. A variable representing a **Files** object.

Parameters

Name	Type	Description
file name0	String	Full path for 1st file (also referred to as source).
file name1	String	Full path for 2nd file (also referred to as target).
NoArrange	Boolean	Optional. Set true if workbooks don't need to be arranged horizontally or vertically.

Return Value

none

8.1.1.3.1.2 Save

Loads the two files from disk.

Syntax

expression.**Save*(Save0,Save1,Path0,Path1)***

expression. A variable representing a **Files** object.

Parameters

Name	Type	Description
Save0	Boolean	Set true if the source file needs to be saved.
Save1	Boolean	Set true if the target file needs to be saved.
Path0	String	Optional. Specifies the path if the source file needs to be saved in a different location.
Path1	String	Optional. Specifies the path if the target file needs to be saved in a different location.

Return Value

none

8.1.1.3.2 Properties

8.1.1.3.2.1 FileName

Returns the full path and file name of either of the Files. **Read only String** value.

Syntax

expression.**file name*(id)***

expression. A variable representing a **Files** object.

Parameters

Name	Type	Description
id	**sideID**	Selects the source or target file.

8.1.1.3.2.2 IsValid

Returns true if both files are loaded and can be compared. **Read only Boolean** value.

Syntax

expression.**IsValid**

expression. A variable representing a **Files** object.

8.1.1.3.2.3 Window

Returns the window object of either of the files. **Read only Window** object.

Syntax

expression.**Window*(id)***
expression. A variable representing a **Files** object.

Parameters

Name	Type	Description
id	**sideID**	Selects source or target file.

8.1.1.3.2.4 Workbook

Returns the workbook object of either of the files. **Read only Workbook** object.

Syntax

expression.**Workbook*(id)***
expression. A variable representing a **Files** object.

Parameters

Name	Type	Description
id	**sideID**	Selects source or target file.

8.1.1.4 Pairs Object

Collection of pair objects.

Methods

Name	Description
AddMatched	Adds matched pairs using MatchType and MatchInclude properties.
AddPair	Adds a pair to the Pairs collection.
Clear	Clears the pairs collection.
Remove	Removes specified pair from the collection.

Properties

Name	Description
Count	Returns the number of members in the collection.
IsValid	Returns true if pairs exist and each pair is valid.
Item	Returns a single item (pair) from the collection.
MatchInclude	Returns or sets options specifying types of worksheets to include by

	AddMatched method.
MatchType	Returns or sets a constant specifying how worksheets will be matched by **AddMatched** method.
PasswordList	Returns or sets a semicolon delimited list of passwords needed to access protected sheets.

Events

Name	Description
none	

Remarks

You can add pairs individually with the **Add** Method or use the **AddMatched** method to automatically add pairs for all worksheets that have matching names or indexes.

You can retrieve a specific pair with Pairs(Index) or Pairs.Item(Index).
You can remove a specific pair with Pairs.Remove(Index).
You can remove all pairs by using the Clear Method.

8.1.1.4.1 Methods

8.1.1.4.1.1 AddMatched

Adds matched pairs using MatchType and MatchInclude properties.

Syntax

*expression.***AddMatched**
expression. A variable representing a **Pairs** object.

Example

This example adds all worksheets with the same name. Protected and hidden sheets are included.

```
With .Pairs
   'include protected and hidden sheets
   .MatchInclude = MatchIncludeFlag_ProtectedSheets + MatchIncludeFlag_HiddenSheets
   'match all sheets with same name
   .MatchType = MatchType_AllByName
   'add sheets
   .AddMatched
End With
```

8.1.1.4.1.2 AddPair

Adds a pair to the Pairs collection.

Syntax

*expression.***AddPair*(Sheet0,Sheet1,Range0,Range1,DBoptions,WSoptions,DBrow,DBkeys)***
expression. A variable representing a **Pairs** object.

Parameters

Name	Type	Description
Sheet0	**String**	A worksheet name of the source file.
Sheet1	**String**	A worksheet name of the target file.
Range0	**String**	A range address of the source worksheet.
Range1	**String**	A range address of the target worksheet.
DBOptions	**DBOptionFlag**	If the worksheet is a database, enter database options here.
WSoptions	**WSOptionFlag**	If the worksheet should be linked 1 on 1, enter linking options.
DBRow	**Long**	If database comparison, enter row with database column/ field names.
DBKeys	**String**	If database comparison, enter the column number(s) which contain(s) the primary key(s). Separate numbers by semicolon(s).

Return Value

Returns a **Pair** object

8.1.1.4.1.3 Clear

Clears the pairs collection.

Syntax

expression.**Clear**
expression. A variable representing a **Pairs** object.

Return Value

none

8.1.1.4.1.4 Remove

Removes specified pair from the collection.

Syntax

expression.**Remove*(index)***
expression. A variable representing a **Pairs** object.

Parameters

Name	Type	Description
index	Long	A valid ID or index for the pair.

Return Value

none

8.1.1.4.2 Properties

8.1.1.4.2.1 Count

Returns the number of members in the collection. **Read only long** value.

Syntax

expression.**Count**

expression. A variable representing a **Pairs** object.

8.1.1.4.2.2 IsValid

Returns true if pairs exist and each pair is valid. **Read only Boolean** value.

Syntax

expression.**IsValid**

expression. A variable representing a **Pairs** object.

8.1.1.4.2.3 Item

Returns a single item (Pair) from the collection. **Pair** object.

Syntax

expression.**Item*(index)***

expression. A variable representing a **Pairs** object.

Parameters

Name	Type	Description
index	Long	A valid ID or index for the pair.

8.1.1.4.2.4 MatchInclude

Returns or sets options specifying types of worksheets to include by **AddMatched** method. **MatchIncludeFlag** constant.

Syntax

expression.**MatchInclude**

expression. A variable representing a **Pairs** object.

Example

This example adds all worksheets with the same name. Protected and hidden sheets are included.

```
With .Pairs
   'include protected and hidden sheets
   .MatchInclude = MatchIncludeFlag_ProtectedSheets + MatchIncludeFlag_HiddenSheets
```

```
    'match all sheets with same name
    .MatchType = MatchType_AllByName
    'add sheets
    .AddMatched
End With
```

8.1.1.4.2.5 MatchType

Returns or sets a variable specifying how worksheets will be matched by **AddMatched** method. **MatchType** constant.

Syntax

*expression.***MatchType**

expression. A variable representing a **Pairs** object.

Example

This example adds all worksheets with equal names. Protected and hidden sheets are included.

```
With .Pairs
    'include protected and hidden sheets
    .MatchInclude = MatchIncludeFlag_ProtectedSheets + MatchIncludeFlag_HiddenSheets
    'match all sheets with same name
    .MatchType = MatchType_AllByName
    'add sheets
    .AddMatched
End With
```

8.1.1.4.2.6 PasswordList

Returns or sets a semicolon delimited list of passwords needed to access protected sheets. **String** value.

Syntax

*expression.***PasswordList**

expression. A variable representing a **Pairs** object.

Example

This example adds two password protected worksheet pairs.

```
With .Pairs
    'add worksheets
    .AddPair "Customer List", "Customer List"
    .AddPair "Customer Profile", "Customer Profile"

    'enter passwords (separated by semicolons)
    .PasswordList = "abc;def"
End With
```

8.1.1.5 Pair Object

Container for a pair of worksheets to compare.

Methods

Name	Description
Activate	Activates the pair object.

Properties

Name	Description
DBKeys	Returns or sets a semicolon delimited list of column indices used as primary key for database.
DBOptions	Returns or sets options for sorting or grouping database records.
DBRow	Returns or sets the index for the row which contains the headers / field names.
IsValid	Returns true if the pair is valid.
Range	Returns the range object for the specified member of the pair.
RangeAddr	Returns or sets an optional range address for the specified member of the pair.
Results	Returns the number of differences found.
Sheet	Returns the worksheet object for the specified member of the pair.
SheetName	Returns the name of the worksheet for the specified member of the pair.
WSOptions	Returns the row/column linking options.

Remarks

Specify a Range
Normally, the entire data range of the sheet is used, but you can limit the range by using the **RangeAddr** property.

Compare as Database
You can compare the sheets as databases if
· your data is organized in a tabular layout, with descriptive headers above each column.
· rows can be uniquely identified by a primary key.

You must set the primary key using the **DBKeys** property.
You can specify the starting row of the database with the **DBRow** property.
If you want the records sorted or the results grouped, use **DBOptions** .

8.1.1.5.1 Methods

8.1.1.5.1.1 Activate

Activates the pair.

Syntax

expression.**Activate**

expression. A variable representing a **Pair** object.

8.1.1.5.2 Properties

8.1.1.5.2.1 DBKeys

Returns or sets a semicolon delimited list of column indices used as primary key for database. **String** value.

Syntax

expression.**DBKeys**
expression. A variable representing a **Pair** object.

Example

This example groups the records.

```
With oProj.Pairs.Item(0)
    'row number of database heading
    .DBRow = 3
    'create primary key; column no. of source file
    .DBKeys = "2;3"
    'group records
    .DBoptions = DBOptionFlag_Group
End With
```

8.1.1.5.2.2 DBOptions

Returns or sets options for sorting or grouping database records. **DBOptionFlag** value.

Syntax

expression.**DBOptions**
expression. A variable representing a **Pair** object.

Example

This example groups the records.

```
With oProj.Pairs.Item(0)
    'row number of database heading
    .DBRow = 3
    'create primary key; column no. of source file
    .DBKeys = "2;3"
    'group records
    .DBoptions = DBOptionFlag_Group
End With
```

8.1.1.5.2.3 DBRow

Returns or sets the index for the row which contains the headers. **Long** value.

Syntax

*expression.***DBRow**
expression. A variable representing a **Pair** object.

Example

This example defines the 3rd row as database heading.

```
With oProj.Pairs.Item(0)
    'row number of database heading
    .DBRow = 3
    'create primary key; column no. of source file
    .DBKeys = "2;3"
    'group records
    .DBoptions = DBOptionFlag_Group
End With
```

8.1.1.5.2.4 IsValid

Returns true if the pair is valid. **Read only Boolean** object.

Syntax

*expression.****IsValid***
expression. A variable representing a **Pair** object.

8.1.1.5.2.5 Range

Returns the range object for the specified member of the pair. **Read only Range** object.

Syntax

*expression.***Range*****(id)***
expression. A variable representing a **Pair** object.

Parameters

Name	Type	Description
id	**sideID**	Selects source or target worksheet.

Remarks

To change the Range use the **RangeAddr** property.

8.1.1.5.2.6 RangeAddr

Returns or sets an optional range address for the specified member of the pair. **Read only String** value.

Syntax

expression.**RangeAddr*(id)***
expression. A variable representing a Pair object.

Parameters

Name	Type	Description
id	sideID	Selects source or target worksheet.

Example

This example sets the range address.

```
With .Pairs
   .Add "db1", "db1"
   .Item(1).RangeAddr(sideID_src) = "A1:H50"
   .Item(1).RangeAddr(sideID_tgt) = "A1:H50"
End With
```

8.1.1.5.2.7 Sheet

Returns the Worksheet object for the specified member of the pair. **Read only Worksheet** object.

Syntax

expression.**Sheet*(id)***
expression. A variable representing a Pair object.

Parameters

Name	Type	Description
id	sideID	Selects source or target worksheet.

8.1.1.5.2.8 SheetName

Returns the name of the worksheet for the specified member of the pair. **Read only String** value.

Syntax

expression.**SheetName*(id)***
expression. A variable representing a Pair object.

Parameters

Name	Type	Description
id	sideID	Selects source or target worksheet.

8.1.1.5.2.9 WSOptions

Returns or sets the name of the linking options.

Syntax

expression.**WSOptions**

expression. A variable representing a **WSOptions** object.

Example

This example links the rows and columns '1 on 1'.

```
With oProj.Pairs.Item(0)
   .WSOptions = WSOptionFlag_Rows1on1 + WSOptionFlag_Cols1on1
End With
```

8.1.1.6 Settings Object

Container for a group of properties that specify how files will be compared and what actions will be performed.

Methods

Name	Description
Clear	Clears all settings.

Properties

Name	Description
Contents	Return or sets the additional content to be compared.
FilterEquivalents	Returns or sets the equivalent values which will be ignored.
Filters	Returns or sets a series of options specifying which differences to filter (ignore).
FilterTolerance	Returns or sets a number below which numeric differences will be ignored.
Formats	Returns or sets a series of options specifying which formatting properties to compare.
Highlight	Returns or sets a variable specifying which highlighting to perform.
Outline	Returns or sets a series of options specifying which group of rows and columns to show or hide.
Report	Returns or sets a variable specifying which report to create.

Events

Name	Description

none

8.1.1.6.1 Methods

8.1.1.6.1.1 Clear

Clears all settings.

Syntax

expression.**Clear**
expression. A variable representing a **Settings** object.

Return Value

none

8.1.1.6.2 Properties

8.1.1.6.2.1 Contents

Returns or sets a series of options specifying which formatting properties to compare. **Contents** constant.

Syntax

expression.**Contents**
expression. A variable representing a **Settings** object.

Example

This example also compares comments and Excel names.

```
With .Settings
  .Contents = ContentFlag_Comments + ContentFlag_Names
End With
```

8.1.1.6.2.2 FilterEquivalents

Returns or sets the equivalent values. **String** value.

Syntax

expression.**FilterEquivalents**
expression. A variable representing a **Settings** object.

Example

This example compares the worksheets using filters.

```
With .Settings
  .Filters = FilterFlag_Enabled + FilterFlag_StringCase + FilterFlag_StringSpace
  .FilterTolerance = 0.01
  .FilterEquivalents = "yes,ja;no,nein"
End With
```

8.1.1.6.2.3 Filters

Returns or sets a series of options specifying which differences to filter (ignore). **FiltersFlag** constant.

Syntax

*expression.***Filters**

expression. A variable representing a **Settings** object.

Remarks

If the ENABLED flag is not set, all other flags are ignored.

Example

This example compares the worksheets using filters.

```
With .Settings
  .Filters = FilterFlag_Enabled + FilterFlag_StringCase + FilterFlag_StringSpace
  .FilterTolerance = 0.01
  .FilterEquivalents = "yes,ja;no,nein"
End With
```

8.1.1.6.2.4 FilterTolerance

Returns or sets a number below which numeric differences will be ignored. **Double** value.

Syntax

*expression.***FilterTolerance**

expression. A variable representing a **Settings** object.

Example

This example compares the worksheets using filters.

```
With .Settings
  .Filters = FilterFlag_Enabled + FilterFlag_StringCase + FilterFlag_StringSpace
  .FilterTolerance = 0.01
  .FilterEquivalents = "yes,ja;no,nein"
End With
```

8.1.1.6.2.5 Formats

Returns or sets a series of options specifying which formatting properties to compare. **FormatsFlag** constant.

Syntax

*expression.***Formats**

expression. A variable representing a **Settings** object.

Remarks

If the ENABLED flag is not set, all other flags are ignored.

Example

This example compares font formats.

```
With .Settings
  .Formats = FormatFlag_Enabled + FormatFlag_Font
End With
```

8.1.1.6.2.6 Highlight

Returns or sets a constant specifying which Highlighting to perform. Uses **HighlightType** constants.

Syntax

expression.**Highlight**
expression. A variable representing a **Settings** object.

Example

This example highlights the differences.

```
With .Settings
  .Highlight = HighlightType_Standard
End With
```

8.1.1.6.2.7 Outline

Returns or sets a series of options specifying which group of Rows and Columns to Show or Hide. Uses **OutlineFlag** constants.

Syntax

expression.**Outline**
expression. A variable representing a **Settings** object.

Remarks

If the ENABLED flag is not set, all other flags are ignored.

Example

This example shows only the differences. Identical rows are hidden.

```
With .Settings
  .Outline = OutlineFlag_Enabled + _
             OutlineFlag_DifferentRows + OutlineFlag_MissingRows + OutlineFlag_Dupl
             OutlineFlag_MissingCols
End With
```

8.1.1.6.2.8 Report

Returns or sets a constant specifying which Report to create. Uses **ReportType** constants.

Syntax

expression.**Report**

expression. A variable representing a **Settings** object.

Example

This example creates a difference report.

```
With .Settings
  .Report = ReportType_Standard
End With
```

8.1.1.7 Results Object

Container for the number of differences found.

Methods

Name	Description
none	

Properties

Name	Description
ArrayCounts	Returns an array with the overall differences found.
ArrayDetails	Returns an array with the individual differences found.
DifferentComments	Returns the number of cells with cell comments.
DifferentContents	Returns the number of cells with different (entered) values and formulas.
DifferentFormats	Returns the number of cells with different formats.
DifferentNames	Returns the number of different Excel names.
DifferentValues	Returns the number of cells with different calculated values (or results).
DuplicateKeys	Returns the number of duplicate keys (database only).
DuplicateRecords	Returns the number of duplicate records (database only).
MissingCols	Returns the number of missing columns (or fields).
MissingRows	Returns the number of missing rows (or records).
MissingSheets	Returns the number of missing worksheets (only for projects with automatching).
Sum	Returns the total number of differences found.
SumByType	Returns the number of specified differences found.
SumText	Returns the results as a formatted text for message box.
SumTextByType	Returns the number of specified differences found as text.

Events

Name	Description

none

Remarks

- If the results object is a child of the **Project** object, then combined results are returned.
- If the results object is a child of the **Pair** object, then individual results are returned.

8.1.1.7.1 Properties

8.1.1.7.1.1 ArrayCounts

Returns an array with the overall number of differences found as text. **Read only Array**.

Syntax

expression.**ArrayCounts**
expression. A variable representing a **Results** object.

Description of Array

This property returns an array. Each difference is represented by a single element of the array with 11
columns. The first element of the array contains the column headings.

Column #	Name	Description
0	Source	Name of source worksheet.
1	Total	Total differences of worksheet pair.
2	Columns	Overall differences of missing columns (source¦target).
3	DupRecs	Overall differences of duplicate records.
4	DupKeys	Overall differences of duplicate keys.
5	Rows	Overall differences of missing rows (source¦target).
6	Content	Overall differences of entered values/formulas.
7	Values	Overall differences of calculated values.
8	Formats	Overall differences of format differences.
9	Comments	Overall differences of cell comments.
10	Names	Overall differences of cell Excel names.

8.1.1.7.1.2 ArrayDetails

Returns an array with the detail information of the specified differences found as text. **Read only Array**.

Syntax

expression.**ArrayDetails*(Type)***
expression. A variable representing a **Results** object.

Parameters

Name	Type	Description
Type	**ResultType**	Constant selects group of results to return. If not set, all result types are returned.

Description of Array

This property returns an array. Each difference is is represented by a single element of the array with 11
columns. The first element of the array contains the column headings.

Column #	Name	Description
0	Source	Name of source worksheet.
1	Adr1	Range address of source worksheet
2	Adr2	Range address of target worksheet
3	Val1	Cell value/formula of source worksheet
4	Val2	Cell value/formula of target worksheet
5	Key	Primary key (only if database)
6	Fld	Field name (only if database)
7	TypeName	Type of difference. The following difference types are displayed: - MissingCol (different columns) - DuplicateRec (duplicate records) - DuplicateKey (duplicate keys) - MissingRow (different rows) - DifferentContent (different entered values/formulas) - DifferentValue (different calculated formulas/results) - DifferentFormat (different formats) - DifferentComment (different cell comments) - DifferentName (different Excel names)
8	TypeSort	Type number of difference.
9	ColSort	Column number of difference.
10	RowSort	Row number of difference.

8.1.1.7.1.3 DifferentComments

Returns the number of different cell comments. **Read only Long** value.

Syntax

*expression.***DifferentComments**
expression. A variable representing a **Results** object.

8.1.1.7.1.4 DifferentContents

Returns the number of cells with different (entered) vales and/or different (entered) formulas. **Read only Long** value.

Syntax

*expression.***DifferentContents**
expression. A variable representing a **Results** object.

8.1.1.7.1.5 DifferentFormats

Returns the number of cells with different formats. **Read only Long** value.

Syntax

*expression.***DifferentFormats**
expression. A variable representing a **Results** object.

8.1.1.7.1.6 DifferentNames

Returns the number of different Excel names. **Read only Long** value.

Syntax

*expression.***DifferentNames**
expression. A variable representing a **Results** object.

8.1.1.7.1.7 DifferentValues

Returns the number of cells with different cell results. **Read only Long** value.

Syntax

*expression.***DifferentValues**
expression. A variable representing a **Results** object.

8.1.1.7.1.8 DuplicateKeys

Returns the number of duplicate keys (database only). **Read only Long** value.

Syntax

*expression.***DuplicateKeys*(id)***
expression. A variable representing a **Results** object.

Parameters

Name	Type	Description
id	**sideID**	Selects either member of the **Pair**

8.1.1.7.1.9 DuplicateRecords

Returns the number of duplicate records (database only). **Read only Long** value.

Syntax

expression.**DuplicateRecords*(id)***
expression. A variable representing a **Results** object.

Parameters

Name	Type	Description
id	**sideID**	Selects either member of the **Pair**

8.1.1.7.1.10 MissingCols

Returns the number of missing columns (or fields). **Read only Long** value.

Syntax

expression.**MissingCols*(id)***
expression. A variable representing a **Results** object.

Parameters

Name	Type	Description
id	**sideID**	Selects either member of the **Pair**

8.1.1.7.1.11 MissingRows

Returns the number of missing rows (or records). **Read only Long** value.

Syntax

expression.**MissingRows*(id)***
expression. A variable representing a **Results** object.

Parameters

Name	Type	Description
id	**sideID**	Selects either member of the **Pair**

8.1.1.7.1.12 MissingSheets

Returns the number of missing worksheets (auto matching only). **Read only Long** value.

Syntax

expression.**MissingSheets*(id)***
expression. A variable representing a **Results** object.

Parameters

Name	Type	Description
id	**sideID**	Selects either member of the **Pair**

8.1.1.7.1.13 Sum

Returns the total number of differences found. **Read only Long** value.

Syntax

*expression.***Sum**
expression. A variable representing a **Results** object.

8.1.1.7.1.14 SumByType

Returns the number of specified differences found. **Read only Long** value.

Syntax

*expression.***SumByType*(Type,id)***
expression. A variable representing a **Results** object.

Parameters

Name	Type	Description
Type	**ResultType**	Constant selects group of results to return.
id	**sideID**	Optional. Only needed for MissingCols and MissingRows. Leave blank to return combined results.

Example

This example prints out some differences.

```
With oProj.Results
  'print out entered value/formula differences
  Debug.Print "Different values/formulas: " & .SumByType(ResultType_DifferentConter

  'print out missing rows in source file
  Debug.Print "Missing Rows Source: " & .SumByType(ResultType_MissingRow, sideID_sr

  'print out missing rows in target file
  Debug.Print "Missing Rows Target: " & .SumByType(ResultType_MissingRow, sideID_tg
End With
```

8.1.1.7.1.15 SumText

Returns a formatted text that can be used in a messagebox. **Read only String** value.

Syntax

*expression.***SummaryText**
expression. A variable representing a **Results** object.

8.1.1.7.1.16 SumTextByType

Returns the number of specified differences found as text. **Read only String** value.

Syntax

expression.**SumTextByType*(Type)***
expression. A variable representing a **Results** object.

Parameters

Name	Type	Description
Type	**ResultType**	Constant selects group of results to return.

8.1.1.8 Enumerations

8.1.1.8.1 Content Flag

Constants used to by the **Contents** property in Settings.

Name	Description
ContentFlag_Comments	If this flag is set, the comments will be compared.
ContentFlag_Names	If this flag is set, the Excel names will be compared.
ContentFlag_None	No additional content will be compared.

8.1.1.8.2 DBOptionFlag

Flags for pair's **DBoptions**

Name	Description
DBOptionFlag_AddID	Adds an ID in the last column of the database to aid sorting and filtering.
DBOptionFlag_Group	Records are grouped by result after comparing.
DBOptionFlag_RelaxedKeys	When selecting this option, primary key differences in spaces and upper/lower case are ignored.
DBOptionFlag_Sort	Records are sorted (on fields in primary key) before comparing.

8.1.1.8.3 FilterFlag

Constants used to by the **Filters** property in Settings.

Name	Description
FilterFlag_Constants	All constants are ignored. Only formulas are compared.

FilterFlag_Datatype	Differences in datatype are ignored. Use to ignore differences between date values and string values representing dates.
FilterFlag_Enabled	Enable or disable all other filters options.
FilterFlag_Formulas	All formulas are ignored. Only constants are compared.
FilterFlag_HiddenCols	Data in hidden columns will be ignored.
FilterFlag_HiddenRows	Data in hidden rows will be ignored.
FilterFlag_None	No filters will be used.
FilterFlag_SameValues	Different cell values/formulas with same result will be ignored.
FilterFlag_StringCase	Text compare case Insensitive. "ABC" vs "abc" will be considered as equal.
FilterFlag_StringSpace	Text compare trim text. "ABC " vs "ABC" will be considered as equal.

8.1.1.8.4 FormatFlag

Constants to be used for the **Formats** property in Settings.

Name	Description
FormatFlag_Alignment	Compare cell alignment properties.
FormatFlag_Border	Compare cell border properties.
FormatFlag_Enabled	Enable or disable all other formats options.
FormatFlag_Font	Compare cell font properties.
FormatFlag_None	No formats are compared.
FormatFlag_Number	Compare cell number (numberformat) properties.
FormatFlag_Pattern	Compare cell pattern (interior/fill) properties.
FormatFlag_Protection	Compare cell protection properties.

8.1.1.8.5 HighlightType

Constants to be used for the **Highlight** property in Settings.

Name	Description
HighlightType_None	Differences are not highlighted.
HighlightType_Standard	Differences are highlighted. No Cell coloring for all cells if not removed before processing.
HighlightType_WithReset	Differences are highlighted. Cell coloring for all cells is removed before processing.

8.1.1.8.6 MatchIncludeFlag

Flags specifying a group of sheets to include during Matching.

Name	Description
MatchIncludeFlag_EmptySheets	Empty sheets will be included by **AddMatched**
MatchIncludeFlag_HiddenSheets	Hidden sheets will be included by **AddMatched**
MatchIncludeFlag_None	Visible worksheets with content will be included. Default setting.
MatchIncludeFlag_ProtectedSheets	Protected sheets will be included by **AddMatched**

8.1.1.8.7 MatchType

List of valid types of matching.

Name	Description
MatchType_AllByCodeName	All included sheets are matched by code name.
MatchType_AllByIndex	All included sheets are matched by index.
MatchType_AllByName	All included sheets are matched by name.
MatchType_FirstByCodeName	The first matched sheets with the same code name are added.
MatchType_FirstByIndex	The first matched sheets are added.
MatchType_FirstByName	The first matched sheets with the same name are added.
MatchType_Manual	Pairs were modified or added with **Add**

8.1.1.8.8 OutlineFlag

Constants used to by the **Outline** property in Settings.

Name	Description
OutlineFlag_DifferentRows	Hide different column: (linked, some cells are different).
OutlineFlag_DuplicateRows	Hide rows with either duplicate keys or records (only in database mode).
OutlineFlag_Enabled	Enable or Disable all other Outline options.
OutlineFlag_IdenticalRows	Hide identical rows (linked, without cell differences).
OutlineFlag_MissingCols	Hide missing columns: (not occurring in the other worksheet).
OutlineFlag_MissingRows	Hide missing rows: (not occurring in the other worksheet).
OutlineFlag_None	No rows/columns will be hidden.

8.1.1.8.9 ReportType

Constants used by the **Report** property in Settings.

Name	Description
ReportType_Hyperlinks	A workbook is created, range addresses are hyperlinks to the original

	document.
ReportType_None	No report is created.
ReportType_Standard	A workbook is created, range addresses are plain text.

8.1.1.8.10 ResultType

Constants to be used for the **ResultType** property in Settings.

Name	Description
ResultType_DifferentComment	Number or detail information of different comments.
ResultType_DifferentContent	Number of different (entered) cell values/formulas.
ResultType_DifferentFormat	Number of different formats.
ResultType_DifferentName	Number of different Excel names.
ResultType_DifferentValue	Number of different calculated cell values (results).
ResultType_DuplicateKey	Number of duplicated primary keys.
ResultType_DuplicateRec	Number of duplicated or redundant records.
ResultType_MissingCol	Number of different columns.
ResultType_MissingRow	Number of different rows.
ResultType_MissingWks	Number of missing worksheets.

8.1.1.8.11 sideID

Constants to identity the member of a set or pair.

Name	Description
sideID_src	Selects the 1st member (Source) of the set. NOTE: value is **0**).
sideID_tgt	Selects the 2nd member (Target) of the set. NOTE: value is **1**).

8.1.1.8.12 WSOptionFlag

Constants used by the **WSOptionFlag** property in Pair.

Name	Description
WSOptionFlag_Cols1on1	Columns will be linked 1 on 1 without finding inserted/deleted columns.
WSOptionFlag_None	Linking is disabled.
WSOptionFlag_Rows1on1	Rows will be linked 1 on 1 without finding inserted/deleted rows.

8.1.2 VBA Helper Functions

```
Option Explicit


'******************************************************************
'*                                                                *
'*                    SYNKRONIZER 11                              *
'*              HELPER PROCEDURES/FUNCTIONS                       *
'*                                                                *
'*  - requires a DEVELOPER license of Synkronizer 11              *
'*  - requires a reference to 'Synkronizer 11 Object Library      *
'*                                                                *
'******************************************************************


Public snk As Synkronizer.Application


'--------------------------------------------------------------
'this wrapper procedure initializes the Synkronizer application object

Public Sub InitSnk(Optional bWithUI As Boolean = False)

  If snk Is Nothing Then
    Dim cai As COMAddIn
    Set cai = Application.COMAddIns("Synkronizer.Addin")
    If Not cai.Connect Then
      'see documentation if following throws permission errors

      'Note:
      'During setup the Addin is installed for All Users by writing
      'the instructions to the registry under
      'HKLM\Software\Microsoft\Office\Excel\Addins

      'Your code should always test that the ComAddin.
      'Connect property returns TRUE.

      'If it returns FALSE, then you can activate the addin
      'from Comaddin Dialog: If you want your code to change
      'the ComAddin.Connect property, then it must be loaded
      'from HKCU (Current user). To make the necessary registry
      'changes we've provided the AddinLoad.bat batch file
      'in the installation folder.

      cai.Connect = True
    End If
    Set snk = cai.Object
  End If

  'Ensure any existing project is silently closed
  If Not bWithUI And snk.DisplayUI Then
```

```
    'hiding the UI will dispose the project
    snk.DisplayUI = False
  ElseIf bWithUI And Not snk.DisplayUI Then
    snk.DisplayUI = True
  End If

End Sub


'------------------------------------------------------------
'this procedure closes a project

Public Sub CloseProject()
  If Not snk.ActiveProject Is Nothing Then
    If Not snk.ActiveProject.IsDisposed Then
      snk.ActiveProject.Close CloseFiles:=False, DisplayUndo:=False
    End If
  End If
End Sub



'----------------------------------------------------
'this procedure checks if the defined
'folders & files are valid

Public Sub Check_Folders_File()

  'check folders
  If ROOT <> "" Then Debug.Assert Len(Dir(ROOT, vbDirectory)) > 0
  If FOLDERSRC <> "" Then Debug.Assert Len(Dir(FOLDERSRC, vbDirectory)) > 0
  If FOLDERTGT <> "" Then Debug.Assert Len(Dir(FOLDERTGT, vbDirectory)) > 0
  If FOLDERREP <> "" Then Debug.Assert Len(Dir(FOLDERREP, vbDirectory)) > 0
  If FOLDERPRJ <> "" Then Debug.Assert Len(Dir(FOLDERPRJ, vbDirectory)) > 0
  If FOLDERLOG <> "" Then Debug.Assert Len(Dir(FOLDERLOG, vbDirectory)) > 0

  'check files
  If FILESRC <> "" Then Debug.Assert Len(Dir(FILESRC)) > 0
  If FILETGT <> "" Then Debug.Assert Len(Dir(FILETGT)) > 0
  If PROTSRC <> "" Then Debug.Assert Len(Dir(PROTSRC)) > 0
  If PROTTGT <> "" Then Debug.Assert Len(Dir(PROTTGT)) > 0

End Sub


'---------------------------------------------------------------
'this function returns the not matched files

Public Function Get_NotMatchedWorksheets(sSrcFolder As String,
sTgtFolder As String) As Variant
  ReDim aFiles(0) As String
  ReDim aNotMatched(0) As String
```

```
  Dim sFile As String
  Dim i, j As Integer

  For i = 1 To 2

    sFile = Dir(Choose(3 - i, sSrcFolder, sTgtFolder) & "*.xls*")
    Do While Len(sFile) > 0
      ReDim Preserve aFiles(UBound(aFiles) + 1)
      aFiles(UBound(aFiles)) = sFile
      sFile = Dir
    Loop

    For j = 1 To UBound(aFiles)
      If Len(Dir(Choose(i, sSrcFolder, sTgtFolder) & aFiles(j))) = 0
Then
        ReDim Preserve aNotMatched(UBound(aNotMatched) + 1)
        aNotMatched(UBound(aNotMatched)) = Choose(i, sSrcFolder,
sTgtFolder) & aFiles(j)
      End If
    Next j

    ReDim aFiles(0)
  Next i

  Get_NotMatchedWorksheets = aNotMatched

End Function


'---------------------------------------------------------------
'this function returns the differences of a project
'if on pair is compared, all detailed differences are returned
'if multiple pairs are compared, the total differences per pair are
returned

Public Function GetDifferences(oProj As Project) As String
  Dim p As Pair
  Dim sMsg As String
  Dim i As Integer

  If oProj.Pairs.Count = 1 Then
    'one pair » return detailed differences
    sMsg = oProj.Results.SumText

  Else
    'multiple pairs » return total differences per pair
    i = 1
    For Each p In oProj.Pairs
      sMsg = sMsg & p.SheetName(sideID_src) & vbTab & _
             p.Results.Sum & vbNewLine
      i = i + 1
      If i > 20 Then Exit For
    Next p
```

```
    'just display the first 20 pairs...
    If i > 20 Then
      sMsg = sMsg & "..." & vbNewLine
    End If
  End If

  GetDifferences = sMsg
End Function


'-----------------------------------------------------------------
'this function compares the files of two folders
'
'Parameter description
'Paramater description:
'sFolderSrc:  Folder with source files to be compared
'sFolderTgt:  Folder with target files to be compared
'bHighlight:  Select True if differences should be highlighted
'sFolderLog:  If difference reports are needed, enter folder. Optional.
'sFolderLog:  If a log file is needed, enter folder. Optional

Public Function SynkFolders(sFolderSrc As String, _
                            sFolderTgt As String, _
                            bHighlight As Boolean, _
                            Optional sFolderRep As String, _
                            Optional sFolderLog As String) As String

  Dim oProj As Synkronizer.Project
  Dim sFile As String
  Dim aFiles() As String
  Dim i As Integer
  Dim j As Integer
  Dim sFileSrc As String
  Dim sFileTgt As String
  Dim sFileRep As String
  Dim sFileLog As String
  Dim vNotMatchedFiles As Variant
  Dim n(0 To 1) As Long
  Dim t0 As Date

  'check if folders are valid
  Debug.Assert Len(Dir(sFolderSrc, vbDirectory))
  Debug.Assert Len(Dir(sFolderTgt, vbDirectory))
  If sFolderRep <> "" Then Debug.Assert Len(Dir(sFolderRep,
vbDirectory))
  If sFolderLog <> "" Then Debug.Assert Len(Dir(sFolderLog,
vbDirectory))

  t0 = Timer

  On Error GoTo theExit
```

```
  'check if defined constants are valid
  Check_Folders_File

  'get access to the Synkronizer application object
  InitSnk

  'create log file
  If sFolderLog <> "" Then
    sFileLog = sFolderLog & "\synkronizer_log_" & Format(Now, "yyyy-mm-
dd_HH-MM-SS") & ".txt"
    Reset
    Open sFileLog For Output As #1
    Print #1, "Synkronizer Logfile"
    Print #1, "--------------------"
    Print #1, ""
    Print #1, "Date: " & Format(Date, "yyyy-mm-dd")
    Print #1, "Time: " & Format(Time, "hh:nn:ss")
    Print #1, ""
    Print #1, ""
  End If


  'read "source" files
  i = 0
  sFile = Dir(sFolderSrc & "*.xls*")
  Do While Len(sFile) > 0
    ReDim Preserve aFiles(i)
    aFiles(i) = sFile
    i = i + 1
    sFile = Dir
  Loop


  'log not matched worksheets
  vNotMatchedFiles = Get_NotMatchedWorksheets(sFolderSrc, sFolderTgt)
  If UBound(vNotMatchedFiles) > 0 Then
    Print #1, "Not matched files"
    For i = 1 To UBound(vNotMatchedFiles)
      Print #1, vNotMatchedFiles(i)
    Next i
    Print #1, ""
    Print #1, ""
  End If


  'loop all "source" files
  For i = 0 To UBound(aFiles)
    sFileSrc = sFolderSrc & aFiles(i)
    sFileTgt = sFolderTgt & aFiles(i)
    sFileRep = sFolderRep & "Difference Report " & aFiles(i)
    sFileRep = Left(sFileRep, InStrRev(sFileRep, ".") - 1) & ".xlsx"

    'check if "target" is there
```

```
    If Len(Dir(sFileTgt)) > 0 Then

      'create new project
      Set oProj = snk.NewProject

      With oProj
        'load files
        .Files.Load sFileSrc, sFileTgt

        'match all worksheets with same name
        With .Pairs
          .MatchType = MatchType_AllByName
          .MatchInclude = MatchIncludeFlag_HiddenSheets +
MatchIncludeFlag_ProtectedSheets
          .AddMatched
        End With

        'highlight & create report
        With .Settings
          If bHighlight Then .Highlight = HighlightType_Standard
          If sFolderRep <> "" Then .Report = ReportType_Standard
        End With

        'compare!
        .Execute

        'log differences
        If sFolderLog <> "" Then
          'Print #1, aFiles(i) & vbTab & .Results.Sum
          Call Logfile_PrintDiffs(oProj)
        End If

        If .Results.Sum Then
          'if differences found, create report
          n(1) = n(1) + 1

          If sFolderRep <> "" Then
            If Len(Dir(sFileRep)) > 0 Then Kill sFileRep
            With .ReportWorkbook
              .SaveAs file name:=sFileRep
            End With
          End If
        Else
          'no differences noted; close report without saving
          n(0) = n(0) + 1

        End If

        'save files if differences are highlighted
        If bHighlight Then
          If .Files.Workbook(sideID_src).FullName <> sFileSrc Then
            .Files.Workbook(sideID_src).SaveCopyAs sFileSrc
          Else
```

```
            .Files.Workbook(sideID_src).Save
          End If
          If .Files.Workbook(sideID_tgt).FullName <> sFileTgt Then
            .Files.Workbook(sideID_tgt).SaveCopyAs sFileTgt
          Else
            .Files.Workbook(sideID_tgt).Save
          End If
        End If

        .Close CloseFiles:=True, DisplayUndo:=False
        DoEvents
      End With

      Set oProj = Nothing
      DoEvents
    End If
  Next i

  'create end message in log file
  If sFolderLog <> "" Then
    Print #1, ""
    Print #1, "Comparison time: " & Format(Timer - t0, " 00.00\s\")
    Reset
  End If

  'display end message
  SynkFolders = "finished" & vbLf & _
                n(0) & " workbooks without differences" & vbLf & _
                n(1) & " workbooks with differences, see reports"

theExit:
  Reset
  Set oProj = Nothing
  Set snk = Nothing
  Exit Function

theError:
  Dim sErr As String
  sErr = Err.Number & ": " & Err.Description
  On Error Resume Next
  If Not oProj Is Nothing Then
    oProj.Close True, False
  End If

  SynkFolders = sErr
  Resume theExit
End Function


'-----------------------------------------------------------------
'this function compares one source file against all files of a target
folder
'
```

```
'Paramater description:
'sFileSrc:    Source file
'sFolderTgt:  Folder with target files to be compared
'bHighlight:  Select True if differences should be highlighted
'sFolderLog:  If difference reports are needed, enter folder. Optional.
'sFolderLog:  If a log file is needed, enter folder. Optional

Public Function SynkSrcFolder(sFileSrc As String, _
                              sFolderTgt As String, _
                              bHighlight As Boolean, _
                              Optional sFolderRep As String, _
                              Optional sFolderLog As String) As String

  Dim oProj As Synkronizer.Project
  Dim aFiles() As String
  Dim i As Integer
  Dim sFile As String
  Dim sFileTgt As String
  Dim sFileRep As String
  Dim sFileLog As String
  Dim n(0 To 1) As Long
  Dim t0 As Date

  'check if files/folders are valid
  Debug.Assert Len(Dir(sFileSrc, vbDirectory))
  Debug.Assert Len(Dir(sFolderTgt, vbDirectory))
  If sFolderRep <> "" Then Debug.Assert Len(Dir(sFolderRep,
vbDirectory))
  If sFolderLog <> "" Then Debug.Assert Len(Dir(sFolderLog,
vbDirectory))

  t0 = Timer

  On Error GoTo theError

  'check if defined constants are valid
  Check_Folders_File

  'get access to the Synkronizer application object
  InitSnk

  'create log file
  If sFolderLog <> "" Then
    sFileLog = sFolderLog & "\synkronizer_log_" & Format(Now, "yyyy-mm-
dd_HH-MM-SS") & ".txt"
    Reset
    Open sFileLog For Output As #1
    Print #1, "Synkronizer Logfile"
    Print #1, "--------------------"
    Print #1, ""
    Print #1, "Date: " & Format(Date, "yyyy-mm-dd")
    Print #1, "Time: " & Format(Time, "hh:nn:ss")
    Print #1, ""
```

```
    Print #1, ""
  End If


  'read "target" files
  i = 0
  sFile = Dir(sFolderTgt & "*.xls*")
  Do While Len(sFile) > 0
    ReDim Preserve aFiles(i)
    aFiles(i) = sFile
    i = i + 1
    sFile = Dir
  Loop

  'loop all files
  For i = 0 To UBound(aFiles)
    sFileTgt = sFolderTgt & aFiles(i)
    sFileRep = sFolderRep & "Difference Report " & aFiles(i)
    sFileRep = Left(sFileRep, InStrRev(sFileRep, ".") - 1) & ".xlsx"

    'create new project
    Set oProj = snk.NewProject

    With oProj
      'load files
      .Files.Load sFileSrc, sFileTgt

      'match all worksheets with same name
      With .Pairs
        .MatchType = MatchType_AllByName
        .MatchInclude = MatchIncludeFlag_HiddenSheets +
MatchIncludeFlag_ProtectedSheets
        .AddMatched
      End With

      'highlight & create report
      With .Settings
        If bHighlight Then .Highlight = HighlightType_Standard
        If sFolderRep <> "" Then .Report = ReportType_Standard
      End With

      'compare!
      .Execute

      'log differences
      If sFolderLog <> "" Then
        'Print #1, aFiles(i) & vbTab & .Results.Sum
        Call Logfile_PrintDiffs(oProj)
      End If

      If .Results.Sum Then

        'if differences found, create report
```

```
        n(1) = n(1) + 1

        If sFolderRep <> "" Then
          If Len(Dir(sFileRep)) > 0 Then Kill sFileRep
          If Not .ReportWorkbook Is Nothing Then
            With .ReportWorkbook
              .SaveAs file name:=sFileRep
            End With
          End If
        End If
      Else
        'no differences noted; close report without saving
        n(0) = n(0) + 1
      End If

      'save target file if differences are highlighted
      If bHighlight Then

        If .Files.Workbook(sideID_tgt).FullName <> sFileTgt Then
          .Files.Workbook(sideID_tgt).SaveCopyAs sFileTgt
        Else
          .Files.Workbook(sideID_tgt).Save
        End If

        '.Files.Workbook(sideID_tgt).Save
      End If

      .Close CloseFiles:=True, DisplayUndo:=False
      DoEvents
    End With

    Set oProj = Nothing
  Next i

  'return value
  SynkSrcFolder = "finished" & vbLf & _
                  n(0) & " workbooks without differences" & vbLf & _
                  n(1) & " workbooks with differences, see reports"



  'write end message in log file
  If sFolderLog <> "" Then
    Print #1, ""
    Print #1, "Comparison time: " & Format(Timer - t0, " 00.00\s\")
    Reset
  End If


theExit:
  Reset
  Set oProj = Nothing
  Set snk = Nothing
```

```
  Exit Function

theError:
  Dim sErr As String
  sErr = Err.Number & ": " & Err.Description
  On Error Resume Next
  If Not oProj Is Nothing Then
    oProj.Close True, False
  End If

  SynkSrcFolder = sErr
  Resume theExit
End Function


'----------------------------------------------------------------
'this procedure writes all project differences in a new workbook

Public Sub DumpDetails_Project(oProj As Synkronizer.Project)
  Dim wkb As Workbook
  Dim wks As Worksheet
  Dim val As Variant
  Dim rng As Range
  Dim p As Pair
  Dim iWksCount As Integer

  Debug.Assert Not oProj Is Nothing
  Debug.Assert Not oProj.IsDisposed

  'create workbook
  iWksCount = Application.SheetsInNewWorkbook
  Application.SheetsInNewWorkbook = 1
  Set wkb = Workbooks.Add
  Application.SheetsInNewWorkbook = iWksCount

  Set wks = wkb.Worksheets(1)

  'name worksheet
  wks.Name = "Project Differences"

  val = oProj.Results.ArrayDetails
  If wks.UsedRange.Cells.CountLarge = 1 Then
    Set rng = wks.Cells(1)
  Else
    With wks.UsedRange
    Set rng = wks.Cells(.Row + .Rows.Count, .Column)
    End With
  End If

  If IsEmpty(val) Then
    rng.Offset(0, 0).Value = "no diffs"
  Else
    Set rng = rng.Offset(0, 0).Resize(UBound(val, 1) + 1, UBound(val, 2)
```

```
+ 1)
    rng.Clear
    rng.Resize(, 11).NumberFormat = "@"
    rng.Resize(, 4).HorizontalAlignment = xlLeft
    rng.VerticalAlignment = xlTop
    rng.Value2 = val

  End If

  'format range
  With rng
    .Rows(1).Font.Bold = True

    .Columns("I:K").HorizontalAlignment = xlRight

    .Columns("A").ColumnWidth = 20
    .Columns("B:K").ColumnWidth = 8
    .Columns("D:E").ColumnWidth = 32
    .Columns("F:H").ColumnWidth = 18

    .Columns("F:G").EntireColumn.Hidden = True
    For Each p In oProj.Pairs
      If p.DBKeys <> "" Then
        .Columns("F:G").EntireColumn.Hidden = False
        Exit For
      End If
    Next p
  End With

  DoEvents
  Application.ScreenUpdating = True

End Sub

'-----------------------------------------------------------------
'this procedure creates a new workbook and
'writes all pair differences in a separate worksheet

Public Sub DumpDetails_Pairs(oProj As Synkronizer.Project)

  Dim wkb As Workbook
  Dim wks As Worksheet
  Dim p As Pair
  Dim iPair As Integer
  Dim val As Variant
  Dim rng As Range
  Dim iWksCount As Integer

  'check if project is active
  Debug.Assert Not oProj Is Nothing
  Debug.Assert Not oProj.IsDisposed

  'create workbook
```

```
iWksCount = Application.SheetsInNewWorkbook
Application.SheetsInNewWorkbook = oProj.Pairs.Count
Set wkb = Workbooks.Add
Application.SheetsInNewWorkbook = iWksCount

'loop through pairs
iPair = 1
For Each p In oProj.Pairs

  'name worksheet
  Set wks = wkb.Worksheets(iPair)
  wks.Name = p.SheetName(0)

  'get results
  val = p.Results.ArrayDetails
  Set rng = wks.Cells(1)

  'write title
  With rng
    .Value = p.SheetName(0)
    .Font.Size = 12
    .Font.Bold = True
  End With

  'write down differences
  If IsEmpty(val) Then
    'no differences found
    rng.Offset(2, 0).Value = "no diffs"
  Else
    'differences found
    Set rng = rng.Offset(2, 0).Resize(UBound(val, 1) + 1, UBound(val,
2) + 1)
    rng.Clear
    rng.Resize(, 6).NumberFormat = "@"
    rng.Resize(, 2).HorizontalAlignment = xlLeft
    rng.VerticalAlignment = xlTop
    rng.Value2 = val
  End If
  val = Empty

  'format range
  With rng
    .Rows(1).Font.Bold = True

    .Columns("I:K").HorizontalAlignment = xlRight

    .Columns("A").ColumnWidth = 20
    .Columns("B:K").ColumnWidth = 8
    .Columns("D:E").ColumnWidth = 32
    .Columns("F:H").ColumnWidth = 18

    If p.DBKeys = "" Then
      .Columns("F:H").EntireColumn.Hidden = True
```

```
      End If
    End With

    iPair = iPair + 1
  Next p

  DoEvents
  Application.ScreenUpdating = True
End Sub


'-----------------------------------------------------------------
'this procedure creates a log file with all differences

Public Sub Logfile_PrintDiffs(oProj As Synkronizer.Project)
  Dim p As Pair
  Dim sText As String
  Dim sLine As String


  'source file
  sText = "Source File"
  sLine = sText & String(34 - Len(sText), " ") & oProj.Files(sideID_src)
  Print #1, sLine

  'target file
  sText = "Target File"
  sLine = sText & String(34 - Len(sText), " ") & oProj.Files(sideID_tgt)
  Print #1, sLine

  '---------------------------------------------------------------
  'heading row differences

  sText = "Worksheet names"
  sLine = sText & String(32 - Len(sText), " ")
  sLine = sLine & "  Total"
  sLine = sLine & _
          "  Columns" & _
          "  DupRecs" & _
          "  DupKeys" & _
          "     Rows" & _
          " Contents" & _
          "   Values"
  If CBool(oProj.Settings.Formats And FormatFlag_Enabled) Then
    sLine = sLine & "  Formats"
  End If
  If CBool(oProj.Settings.Contents And ContentFlag_Comments) Then
    sLine = sLine & " Comments"
  End If
  If CBool(oProj.Settings.Contents And ContentFlag_Names) Then
    sLine = sLine & "    Names"
  End If
```

```
Print #1, sLine

'---------------------------------------------------------------
'pair differences

For Each p In oProj.Pairs
  With p.Results
    sText = p.SheetName(sideID_src)
    sLine = sText & String(32 - Len(sText), " ")

    sText = CStr(.Sum)
    sLine = sLine & String(7 - Len(sText), " ") & sText

    sText = .SumTextByType(ResultType_MissingCol)
    sLine = sLine & String(9 - Len(sText), " ") & sText

    sText = .SumTextByType(ResultType_DuplicateRec)
    sLine = sLine & String(9 - Len(sText), " ") & sText

    sText = .SumTextByType(ResultType_DuplicateKey)
    sLine = sLine & String(9 - Len(sText), " ") & sText

    sText = .SumTextByType(ResultType_MissingRow)
    sLine = sLine & String(9 - Len(sText), " ") & sText

    sText = .SumTextByType(ResultType_DifferentContent)
    sLine = sLine & String(9 - Len(sText), " ") & sText

    sText = .SumTextByType(ResultType_DifferentValue)
    sLine = sLine & String(9 - Len(sText), " ") & sText

    If CBool(oProj.Settings.Formats And FormatFlag_Enabled) Then
      sText = .SumTextByType(ResultType_DifferentFormat)
      sLine = sLine & String(9 - Len(sText), " ") & sText
    End If

    If CBool(oProj.Settings.Contents And ContentFlag_Comments) Then
      sText = .SumTextByType(ResultType_DifferentComment)
      sLine = sLine & String(9 - Len(sText), " ") & sText
    End If

    If CBool(oProj.Settings.Contents And ContentFlag_Names) Then
      sText = .SumTextByType(ResultType_DifferentName)
      sLine = sLine & String(9 - Len(sText), " ") & sText
    End If

    Print #1, sLine
  End With
Next p


'---------------------------------------------------------------
```

```
'total differences

If oProj.Pairs.Count > 1 Then
  With oProj.Results
    sText = "Total"
    sLine = sText & String(32 - Len(sText), " ")

    sText = CStr(.Sum)
    sLine = sLine & String(7 - Len(sText), " ") & sText

    sText = .SumTextByType(ResultType_MissingCol)
    sLine = sLine & String(9 - Len(sText), " ") & sText

    sText = .SumTextByType(ResultType_DuplicateRec)
    sLine = sLine & String(9 - Len(sText), " ") & sText

    sText = .SumTextByType(ResultType_DuplicateKey)
    sLine = sLine & String(9 - Len(sText), " ") & sText

    sText = .SumTextByType(ResultType_MissingRow)
    sLine = sLine & String(9 - Len(sText), " ") & sText

    sText = .SumTextByType(ResultType_DifferentContent)
    sLine = sLine & String(9 - Len(sText), " ") & sText

    sText = .SumTextByType(ResultType_DifferentValue)
    sLine = sLine & String(9 - Len(sText), " ") & sText

    If CBool(oProj.Settings.Formats And FormatFlag_Enabled) Then
      sText = .SumTextByType(ResultType_DifferentFormat)
      sLine = sLine & String(9 - Len(sText), " ") & sText
    End If

    If CBool(oProj.Settings.Contents And ContentFlag_Comments) Then
      sText = .SumTextByType(ResultType_DifferentComment)
      sLine = sLine & String(9 - Len(sText), " ") & sText
    End If

    If CBool(oProj.Settings.Contents And ContentFlag_Names) Then
      sText = .SumTextByType(ResultType_DifferentName)
      sLine = sLine & String(9 - Len(sText), " ") & sText
    End If

    Print #1, sLine
  End With
End If

Print #1, ""
Print #1, ""

End Sub
```

8.1.3 VBA Examples

```
Option Explicit


'*******************************************************************
'*                                                                 *
'*                    SYNKRONIZER 11                               *
'*                     VBA EXAMPLES                                *
'*                                                                 *
'*  To test the examples you need                                  *
'*  - to have a DEVELOPER license of Synkronizer 11                *
'*  - to create a reference to 'Synkronizer 11 Object Library'     *
'*  - include the module 'snk_helper_functions' which contains    *
'*    helper procedures and functions                              *
'*                                                                 *
'*******************************************************************


'define folders
Public Const ROOT As String = "D:\Documents\"
Public Const FOLDERSRC As String = ROOT & "Source\"
Public Const FOLDERTGT As String = ROOT & "Target\"
Public Const FOLDERREP As String = ROOT & "Reports\"
Public Const FOLDERPRJ As String = ROOT & "Projects\"
Public Const FOLDERLOG As String = ROOT & "Log\"

'define files
Public Const FILESRC As String = FOLDERSRC & "Source 1.xlsx"
Public Const FILETGT As String = FOLDERTGT & "Target 2.xlsx"
Public Const FILEREP As String = FOLDERREP & "Synkronizer_Difference_Report.xlsx"
Public Const FILEPRJ As String = FOLDERPRJ & "SynkProject.xml"
Public Const PROTSRC As String = FOLDERSRC & "Protected 1.xlsx"
Public Const PROTTGT As String = FOLDERTGT & "Protected 2.xlsx"


'------------------------------------------------------------
'Example 1
'- compare all sheets with the same name
'- create a difference report
'- highlight differences
'- show all differences except identical rows
'- create a project with all settings

Public Sub Example1()
  Dim oProj As Synkronizer.Project
  Dim sMsg As String

  On Error GoTo Err_Example

  'check if defined constants are valid
  Check_Folders_File
```

```
'get access to the Synkronizer application object
InitSnk

'create project
Set oProj = snk.NewProject
With oProj

  'load files
  .Files.Load FILESRC, FILETGT

  'define worksheets
  With .Pairs
    .MatchType = MatchType_AllByName
    .AddMatched
  End With

  With .Settings

    'create a report
    .Report = ReportType_Standard

    'highlight differences
    .Highlight = HighlightType_WithReset

    'show only rows/columns with differences
    .Outline = OutlineFlag_Enabled + _
               OutlineFlag_DifferentRows + OutlineFlag_MissingRows + OutlineFlag_
               OutlineFlag_MissingCols

    'also compare comments and names
    .Contents = ContentFlag_Comments + ContentFlag_Names

    'also compare also font formats
    .Formats = FormatFlag_Enabled + FormatFlag_Font

  End With

  'start Synkronizer
  .Execute

  'get the message string before closing the project
  sMsg = GetDifferences(oProj)

  'save project
  If snk.DisplayUI = False Then
    .Save (FILEPRJ)
  End If

  'close project
  CloseProject
```

```
  End With

  'display message
  MsgBox sMsg, vbOKOnly + vbInformation, "Synkronizer"

  Exit Sub

Err_Example:
  MsgBox Err.Description, vbExclamation, "Synkronizer"
End Sub


'------------------------------------------------------------
'Example 2
'- compare a project

Public Sub Example2()
  Dim oProj As Synkronizer.Project
  Dim sMsg As String

  On Error GoTo Err_Example

  'check if defined constants are valid
  Check_Folders_File

  'get access to the Synkronizer application object
  InitSnk

  'load Synkronizer project
  Set oProj = snk.OpenProject(FILEPRJ)

  'check if source & target files are valid
  Debug.Assert oProj.Files.IsValid

  'start Synkronizer
  oProj.Execute

  'get the message string before closing the project
  sMsg = GetDifferences(oProj)

  'close project
  CloseProject

  'display message
  MsgBox sMsg, vbOKOnly + vbInformation, "Synkronizer"

  Exit Sub

Err_Example:
  MsgBox Err.Description, vbExclamation, "Synkronizer"
End Sub
```

```
'-----------------------------------------------------------
'Example 3
'- compare protected sheets
'- highlight differences.

Public Sub Example3()
  Dim oProj As Synkronizer.Project
  Dim sMsg As String

  On Error GoTo Err_Example

  'check if defined constants are valid
  Check_Folders_File

  'get access to the Synkronizer application object
  InitSnk

  'create project
  Set oProj = snk.NewProject

  With oProj

    'define files
    .Files.Load PROTSRC, PROTTGT

    With .Pairs

      'add worksheets
      .AddPair "Customer List", "Customer List"
      .AddPair "Customer Profile", "Customer Profile"

      'enter passwords (separated by semicolons)
      .PasswordList = "abc;def"
   End With

    'highlight differences
    With .Settings
      .Highlight = HighlightType_Standard
    End With

    'start Synkronizer
    .Execute

    'get the message string before closing the project
    sMsg = GetDifferences(oProj)

    'close project
    CloseProject

  End With
```

```
  'display message
  MsgBox sMsg, vbOKOnly + vbInformation, "Synkronizer"

  Exit Sub

Err_Example:
  MsgBox Err.Description, vbExclamation, "Synkronizer"
End Sub


'------------------------------------------------------------
'Example 4
'- compare first sheets
'- compare formats, comments, names & use filters
'- create a report

Public Sub Example4()
  Dim oProj As Synkronizer.Project
  Dim sMsg As String

  On Error GoTo Err_Example

  'check if defined constants are valid
  Check_Folders_File

  'get access to the Synkronizer application object
  InitSnk

  'create project
  Set oProj = snk.NewProject

  With oProj

    'define files
    .Files.Load FILESRC, FILETGT

    'define first worksheet of each file
    With .Pairs
      .MatchType = MatchType_FirstByName
      .AddMatched
    End With

    With .Settings

      'also compare comments and names
      .Contents = ContentFlag_Comments + ContentFlag_Names

      'compare also font formats
      .Formats = FormatFlag_Enabled + FormatFlag_Font

      'use some filters
      .Filters = FilterFlag_Enabled + FilterFlag_StringCase + FilterFlag_StringSpac
```

```
        .FilterTolerance = 0.01
        .FilterEquivalents = "yes,ja;no,nein"

        'create a report
        .Report = ReportType_Standard
      End With

      'compare files
      .Execute

      If Not .ReportWorkbook Is Nothing Then

        'delete report if it already there
        If Len(Dir(FILEREP)) > 0 Then Kill FILEREP

        'save report
        With .ReportWorkbook
          .SaveAs file name:=FILEREP
          .Close SaveChanges:=False
        End With
      End If

      'get the message string before closing the project
      sMsg = GetDifferences(oProj)

      'close project
      CloseProject

    End With

    'display message
    MsgBox sMsg, vbOKOnly + vbInformation, "Synkronizer"

    Exit Sub

Err_Example:
    MsgBox Err.Description, vbExclamation, "Synkronizer"
End Sub


'------------------------------------------------------------
'Example 5
'- database comparison
'- highlight differences

Public Sub Example5()
  Dim oProj As Synkronizer.Project
  Dim sMsg As String

  On Error GoTo Err_Example

  'check if defined constants are valid
```

```
  Check_Folders_File

  'get access to the Synkronizer application object
  InitSnk

  'create project
  Set oProj = snk.NewProject

  With oProj

    'define files
    .Files.Load FILESRC, FILETGT

    'define worksheets & database options
    .Pairs.AddPair Sheet0:="Customer List", _
                   Sheet1:="Customer List", _
                   DBRow:=1, _
                   DBKeys:="1;2", _
                   DBOptions:=DBOptionFlag_Group

    'highlight differences
    With .Settings
      .Highlight = HighlightType_WithReset
    End With

    'start Synkronizer
    .Execute

    'get the message string before closing the project
    sMsg = GetDifferences(oProj)

    'close project
    CloseProject

  End With

  'display message
  MsgBox sMsg, vbOKOnly + vbInformation, "Synkronizer"

  Exit Sub

Err_Example:
  MsgBox Err.Description, vbExclamation, "Synkronizer"
End Sub


'------------------------------------------------------------
'Example 6
'- link rows 1 on 1
'- highlight differences

Public Sub Example6()
```

```
    Dim oProj As Synkronizer.Project
    Dim sMsg As String

    On Error GoTo Err_Example

    'check if defined constants are valid
    Check_Folders_File

    'get access to the Synkronizer application object
    InitSnk

    'create project
    Set oProj = snk.NewProject

    With oProj

      'define files
      .Files.Load FILESRC, FILETGT

      'define worksheets & link options
      .Pairs.AddPair Sheet0:="Customer List", _
                     Sheet1:="Customer List", _
                     WSOptions:=WSOptionFlag_Rows1on1

      'highlight differences
      With .Settings
        .Highlight = HighlightType_WithReset
      End With

      'start Synkronizer
      .Execute

      'get the message string before closing the project
      sMsg = GetDifferences(oProj)

      'close project
      CloseProject

    End With

    'display message
    MsgBox sMsg, vbOKOnly + vbInformation, "Synkronizer"

    Exit Sub

  Err_Example:
    MsgBox Err.Description, vbExclamation, "Synkronizer"
  End Sub


  '------------------------------------------------------------
  'Example 7
```

```
'- compare a project
'- create workbook which contains all differences

Public Sub Example7()
  Dim oProj As Synkronizer.Project
  Dim sMsg As String

  On Error GoTo Err_Example

  'check if defined constants are valid
  Check_Folders_File

  'get access to the Synkronizer application object
  InitSnk

  'load Synkronizer project
  Set oProj = snk.OpenProject(FILEPRJ)

  'check if source & target files are valid
  Debug.Assert oProj.Files.IsValid

  'start Synkronizer
  oProj.Execute

  'create a workbook with all detail differences
  DumpDetails_Project oProj

  'create a workbook with all detail differences
  'all pair differences will be written in a separate worksheet
  DumpDetails_Pairs oProj

  'get the message string before closing the project
  sMsg = GetDifferences(oProj)

  'close project
  CloseProject

  'display message
  MsgBox sMsg, vbOKOnly + vbInformation, "Synkronizer"

  Exit Sub

Err_Example:
  MsgBox Err.Description, vbExclamation, "Synkronizer"
End Sub


'-----------------------------------------------------------
'Example 8
'- compare all Excel files with the same names of two folders
'- a difference report will be created of each file set
'- a log file will be created
```

```
Public Sub Example8()
  Dim sMsg As String

  'Paramater description:
  'sFolderSrc:  Folder with source files to be compared
  'sFolderTgt:  Folder with target files to be compared
  'bHighlight:  Select True if differences should be highlighted
  'sFolderLog:  If difference reports are needed, enter folder. Optional.
  'sFolderLog:  If a log file is needed, enter folder. Optional

  sMsg = SynkFolders(sFolderSrc:=FOLDERSRC, _
                     sFolderTgt:=FOLDERTGT, _
                     bHighlight:=True, _
                     sFolderRep:=FOLDERREP, _
                     sFolderLog:=FOLDERLOG)

  MsgBox sMsg

End Sub


'------------------------------------------------------------
'Example 9
'- compare one "source" file against a series of target files
'- a difference report will be created for each file set
'- a log file will be created

Public Sub Example9()
  Dim sMsg As String

  'Paramater description:
  'sFileSrc:    Source file
  'sFolderTgt:  Folder with target files to be compared
  'bHighlight:  Select True if differences should be highlighted
  'sFolderLog:  If difference reports are needed, enter folder. Optional.
  'sFolderLog:  If a log file is needed, enter folder. Optional

  sMsg = SynkSrcFolder(sFileSrc:=FILESRC, _
                       sFolderTgt:=FOLDERTGT, _
                       bHighlight:=True, _
                       sFolderRep:=FOLDERREP, _
                       sFolderLog:=FOLDERLOG)

  MsgBox sMsg

End Sub
```

8.2 CommandLine Utility

With the Developer Edition of Synkronizer 11, it is also possible to compare Excel files via the **command line commands**. **The Excel files can be compared fully automated with a single mouse click - without opening Excel!**

How to use the CommandLine Utility?

synk.exe is a handy program which allows you to compare Excel files without opening Excel. You can enter all variables like file names, folders and comparison options in a single command. The program then generates different reports and log files with all differences. While executing, the program will start a hidden instance of Excel. After the comparison process the Excel instance will be closed again.

Which tasks does the CommandLine Utility support?

The following tasks are supported:
- Compare two files
- Compare a source file against a bundle of source files
- Compare all Excel files (with equal file names) in two folders
- Create a difference report
- Create a log file which contains all differences
- Format and filter options are supported

Which tasks are not supported by the CommandLine Utility?

The following tasks are not supported:
- Source and target files cannot be saved
- Differences cannot be highlighted
- Differences cannot be outlined (shown/hidden)

The CommandLine Utility of Synkronizer is invoked as follows:

1. Start the Windows Explorer
2. Select the Synkronizer folder
   `c:\Program Files\Synkronizer\Synkronizer 11\synk.exe`
3. Double-click on **synk.exe**
4. The Sykronizer CommandLine Utility appears.

```
Synkronizer 11
Synkronizer CommandLine Utility

Variables
    dir          D:\Documents\
    src
    tgt
    xml
    rep
    log
Tools
    matching     1 (AllByName)
    report       1 (Standard)

=>
```

Start CommandLine Utility with a batch file or Windows shortcut

We recommend that you create a batch file or a Windows shortcut. In this way you can start the CommandLine Utility with a single mouse click.

Creating a batch file

Proceed as follows:

1. Start a text editor and create a new document.
2. Write the following code:

```
@echo off
"C:\Program Files\Synkronizer\Synkronizer 11\Synk.exe"
```

3. Save the file as synk.bat in a folder of preference

Creating a Windows shortcut

You can also create a Windows shortcut. Proceed as follows:

1. Right-click an open area on the desktop, point to New, and then click Shortcut.
2. Click Browse.
3. Locate **synk.exe** which is stored in the following folder:
   `C:\Program Files\Synkronizer\Synkronizer 11\`
   click Open, and then click Next.
4. Type a name for the shortcut and click Finish button.
5. Right-click on the created shortcut.
6. Enter in the field "Start in" the folder, in which the Synkronizer files should be stored.
7. Click OK and close the shortcut

8.2.1 Reference

This page contains the syntax of the command line utility and a description of all variables, commands and settings.

Syntax

```
synk[.exe]
  [/dir={Parent folder}]
  [/src={Source folder/file}]
  [/tgt={Target folder/file}]
  [/xml={Synkronizer project file}]
  [/rep={Report folder}]
  [/log={Log file}]
  [/m={Matching options}]
  [/r={Report options}]
  [/c]
  [/x]
```

Variables

/dir={Parent folder}
Optional. Name/location of the parent folder. The difference reports and log files will be stored within this folder if not defined separately.

/src={Source folder/file}
Required. Folder of the **source files** or the name of the **source file**. If you want to compare all files of a folder you need to enter a folder otherwise enter a file name. You can enter an absolute path/ file or a path/file name relative to the parent directory (/dir).

/tgt={Target folder/file}
Required. Folder of the **target files** or the name of the **target file**. If you want to compare all files of a folder you need to enter a folder otherwise enter a file name. You can enter an absolute path/ file or a path/file name relative to the parent directory (/dir).

/xml={Synkronizer Project file}
Optional. Name of the Synkronizer project file. All other settings like Contents, Formats or Filters are taken from this project file. If no project file is defined, the Synkronizer default settings are used.

/rep={Folder for difference reports}
Optional. Name of the folder in which the difference reports are saved. If this parameter is empty, the difference reports are written into the parent folder (/dir).

/log={Name of Synkronizer log file}
Optional. Name of the log file. If this parameter is defined, a log file will be created.

Note
If the folders or file names contain **spaces** and you are working with **batch files** and/or **Windows shortcuts** you need to use **apostrophes** (double quotes) for addressing the folders/file names.

Tools

matching (/m=)
Optional. Matching options. You can define how the worksheets shall be matched. The following options are allowed:
1=Compare all worksheets by name (default setting)
2=Compare all worksheets by index
-1=Compare first worksheets by name
-2=Compare first worksheets by index

report (/r=)
Optional. Difference report options. The following options are allowed:
1=A standard difference report is created (default setting)
2=A difference with hyperlinks will be created.
0=No difference report will be created.

Commands

/?
Synkronizer CommandLine Utility help. All commands and settings are displayed.

/f
Files to be compared. You can see which files will be compared and which not.

/v
The settings of the current project are displayed. You can see the folders, file names and comparison settings.

/c
Starts the comparison process. You can enter ctrl-c to terminate the comparison process.

/x
The Synkronizer CommandLine Utility will be closed.

8.2.2 Examples

This section contains some examples on how you can compare Excel files with the Synkronizer CommandLine Utility.

Example 1 - Compare two files

1. Start the CommandLine Utility of Synkronizer using **synk.exe**
2. Enter the source file with the command **src={file name}**
3. Enter the target file with the command **tgt={file name}**
4. Compare the files with the command **c**
5. Close the Synkronizer CommandLine Utility with the command **x**

```
Synkronizer 11
Synkronizer CommandLine Utility

Variables
    dir          D:\Documents\
    src
    tgt
    xml
    rep
    log
Tools
    matching     1 (AllByName)
    report       1 (Standard)


=> src=Source\File1.xlsx
=> tgt=Target\File2.xlsx
=> c
```

Direct command for batch file or Windows shortcut:
`"C:\Program Files\Synkronizer\Synkronizer 11\Synk.exe" /dir="D:\Documents\" /src="S`

Example 2 - Compare one master file with various target files

1. Start the CommandLine Utility of Synkronizer using **synk.exe**
2. Enter the source file with the command **src={file name}**
3. Enter the target folder (which contains the target files) with the command **tgt={target folder}**
4. Compare the files with the command **c**
5. Close the Synkronizer CommandLine Utility with the command **x**

```
Synkronizer 11
Synkronizer CommandLine Utility

Variables
    dir          D:\Documents\
    src
    tgt
    xml
    rep
    log
Tools
    matching     1 (AllByName)
    report       1 (Standard)


=> src=Source\File1.xlsx
=> tgt=Target\*.*
=> c
```

Direct command for batch file or Windows shortcut:
`"C:\Program Files\Synkronizer\Synkronizer 11\Synk.exe" /dir="D:\Documents\" /src="S`

Example 3 - Compare all files in two folders

1. Start the CommandLine Utility of Synkronizer using **synk.exe**
2. Enter the source folder with the command **src={source folder}**
3. Enter the target folder with the command **tgt={target folder}**
4. Compare the files with the command **c**
5. Close the Synkronizer CommandLine Utility with the command **x**

```
Synkronizer 11
Synkronizer CommandLine Utility

Variables
    dir           D:\Documents\
    src
    tgt
    xml
    rep
    log
Tools
    matching      1 (AllByName)
    report        1 (Standard)


=> src=Source\*.*
=> tgt=Target\*.*
=> c
```

Direct command for batch file or Windows shortcut:

```
"C:\Program Files\Synkronizer\Synkronizer 11\Synk.exe" /dir="D:\Documents\" /src="S
```

Example 4 - Create no difference report but a logfile

1. Start the CommandLine Utility of Synkronizer using **synk.exe**
2. Enter the source file with the command **src={file name}**
3. Enter the target file with the command **tgt={file name}**
4. Enter the name of the logfile: **log={file name}**
5. Define report settings. Enter **r=0**
6. Compare the files with the command **c**
7. Close the Synkronizer CommandLine Utility with the command **x**

```
Synkronizer 11
Synkronizer CommandLine Utility

Variables
   dir          D:\Documents\
   src
   tgt
   xml
   rep
   log
Tools
   matching     1 (AllByName)
   report       1 (Standard)


=> src=Source\File1.xlsx
=> tgt=Target\File2.xlsx
=> log=synklog.txt
=> r=0
=> c
```

Direct command for batch file or Windows shortcut:
`"C:\Program Files\Synkronizer\Synkronizer 11\Synk.exe" /dir="D:\Documents\" /src="S`

Example 5 - Compare two files using filters and formats

1. Start Excel & Synkronizer
2. Define the files, filters and formats which you'd like to compare
3. Save the settings as a project
4. Close Synkronizer and Excel
5. Start the CommandLine Utility of Synkronizer using **synk.exe**
6. Enter the source file with the command **src={file name}**
7. Enter the target file with the command **tgt={file name}**
8. Enter the Synkronizer project file which contains the settings: **xml={file name}**
9. Compare the files with the command **c**
10. Close the Synkronizer CommandLine Utility with the command **x**

```
Synkronizer 11
Synkronizer CommandLine Utility

Variables
   dir          D:\Documents\
   src
   tgt
   xml
   rep
   log
Tools
   matching     1 (AllByName)
   report       1 (Standard)


=> src=Source\File1.xlsx
=> tgt=Target\File2.xlsx
=> xml=Projects\SynkProject.xml
=> c
```

Direct command for batch file or Windows shortcut:

```
"C:\Program Files\Synkronizer\Synkronizer 11\Synk.exe" /dir="D:\Documents\" /src="S
```

Index

- " -

- A -

- B -

- C -

- D -

- E -

- F -

- G -

- H -

- I -

- K -

- L -

- T -

- U -

- V -

- W -

- X -

- Z -